# User Guides

## 1 Source Guide
- 1.1 Choosing the Right Location .......................................................... 3
- 1.2 Get the Tor Browser ........................................................................ 3
- 1.3 Choose Who to Submit To ................................................................. 4
- 1.4 Making Your First Submission ......................................................... 4
- 1.5 Continuing the Conversation ............................................................ 9

## 2 Journalist Guide
- 2.1 Updating Your Workstation ............................................................... 13
- 2.2 Connecting to the Journalist Interface ............................................. 14
- 2.3 Daily Journalist Alerts About Submissions ....................................... 15
- 2.4 Interacting With Sources ................................................................. 16
- 2.5 Moving Documents to the Secure Viewing Station .......................... 18
- 2.6 Working with Documents ............................................................... 23
- 2.7 Moving Documents to Your Everyday Workstation ......................... 26
- 2.8 Decrypting and Preparing to Publish .............................................. 27

## 3 Admin Guide
- 3.1 Responsibilities .............................................................................. 29
- 3.2 Common Tasks ............................................................................... 31
- 3.3 Frequently Asked Questions ............................................................ 40

## 4 Passphrase Best Practices
- 4.1 General Best Practices ................................................................... 43
- 4.2 For Sources .................................................................................... 44
- 4.3 For Journalists/Admins ................................................................... 44

## 5 Overview
- 5.1 Technical Summary ......................................................................... 45
- 5.2 Infrastructure .................................................................................. 46
- 5.3 Operation ....................................................................................... 47

## 6 Glossary
- 6.1 Source ........................................................................................... 49
- 6.2 Journalist ....................................................................................... 49
- 6.3 Application Server ......................................................................... 49
- 6.4 Monitor Server ............................................................................... 50
16 Set Up the Servers 137
16.1 Install Ubuntu ................................................. 137
16.2 Test Connectivity .............................................. 142
16.3 Set Up SSH Keys .............................................. 142

17 Install SecureDrop 145
17.1 Install Prerequisites ......................................... 145
17.2 Localization of the Source Interface and Journalist Interface . 146
17.3 Onion Service Options ....................................... 146
17.4 Configure the Installation ................................... 147
17.5 Install SecureDrop Servers ................................ 148

18 Configure the Admin Workstation Post-Install and Create Backups 151
18.1 Auto-connect to the Authenticated Tor Onion Services ........................................... 151
18.2 Back Up the Workstations ................................ 152

19 Create an Admin Account on the Journalist Interface 153

20 Test the Installation 155
20.1 Test Connectivity .............................................. 155
20.2 Sanity-Check the Installation ................................ 156
20.3 Test the Web Interfaces ...................................... 156

21 Onboard Journalists 157
21.1 Determine Access Protocol for the Secure Viewing Station ........................................... 157
21.2 Create a Journalist Tails USB ................................ 158
21.3 Set Up Automatic Access to the Journalist Interface ................................................. 158
21.4 Add an account on the Journalist Interface ................................................................. 159
21.5 Provision a personal Transfer Device and Export Device ............................................ 159
21.6 Verify Journalist Setup ........................................ 159

22 Overview 161

23 Landing Page 163
23.1 URL and Location .............................................. 163
23.2 HTTPS Only (No Mixed Content) ................................ 163
23.3 Perfect Forward Secrecy ...................................... 164
23.4 SSL Certificate Recommendations ................................ 164
23.5 Do Not Use Third-Party Analytics, Tracking, or Advertising ...................................... 164
23.6 Do Not Hyperlink .onion Addresses ................................ 165
23.7 Avoid Direct Links to SecureDrop.org ................................ 165
23.8 Apply Security Headers ........................................ 165
23.9 Additional Apache Configuration ................................ 166
23.10 Further Security Considerations ................................ 167
23.11 Landing Page Content Suggestions ................................ 167

24 Minimum requirements for the SecureDrop environment 169

25 Whole Site Changes 171
25.1 Suggested ...................................................... 171

26 Sample SecureDrop Privacy Policy 173
26.1 Collection of Information From Sources ................................................................. 173
26.2 Collection of Information About Journalists’ Use of SecureDrop ................................ 174
26.3 Data Security .................................................... 174
26.4 Children Under 13 ............................................. 174
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>26.5</td>
<td>Changes to This Policy</td>
<td>174</td>
</tr>
<tr>
<td>26.6</td>
<td>Contact</td>
<td>174</td>
</tr>
<tr>
<td>27</td>
<td>Pre-Install Hardware Checklist</td>
<td>175</td>
</tr>
<tr>
<td>28</td>
<td>Promoting Your SecureDrop Instance</td>
<td>177</td>
</tr>
<tr>
<td>28.1</td>
<td>Make a High Profile Announcement</td>
<td>177</td>
</tr>
<tr>
<td>28.2</td>
<td>Provide a Clear Link on Your Homepage</td>
<td>177</td>
</tr>
<tr>
<td>28.3</td>
<td>Provide Links at the Bottom of Your Articles</td>
<td>179</td>
</tr>
<tr>
<td>28.4</td>
<td>Create an Instructional Video on How to Access and Use Your SecureDrop</td>
<td>179</td>
</tr>
<tr>
<td>28.5</td>
<td>Regularly Share Your SecureDrop Landing Page on Social Media</td>
<td>179</td>
</tr>
<tr>
<td>28.6</td>
<td>Target Potential Whistleblowers with Advertising</td>
<td>180</td>
</tr>
<tr>
<td>28.7</td>
<td>Put an Advertisement in Your Physical Paper</td>
<td>180</td>
</tr>
<tr>
<td>29</td>
<td>What Makes SecureDrop Unique</td>
<td>183</td>
</tr>
<tr>
<td>29.1</td>
<td>No Third Parties that Can Secretly be Subpoenaed</td>
<td>183</td>
</tr>
<tr>
<td>29.2</td>
<td>Limits the Metadata Trail as Much as Possible</td>
<td>183</td>
</tr>
<tr>
<td>29.3</td>
<td>Encrypted and Air-Gapped</td>
<td>184</td>
</tr>
<tr>
<td>29.4</td>
<td>Protects Against Hackers</td>
<td>184</td>
</tr>
<tr>
<td>29.5</td>
<td>Free and Open Source Software</td>
<td>184</td>
</tr>
<tr>
<td>30</td>
<td>Investigating Logs</td>
<td>185</td>
</tr>
<tr>
<td>30.1</td>
<td>Logs to examine on both servers</td>
<td>185</td>
</tr>
<tr>
<td>30.2</td>
<td>Application Server Logs</td>
<td>185</td>
</tr>
<tr>
<td>30.3</td>
<td>Monitor Server Logs</td>
<td>186</td>
</tr>
<tr>
<td>31</td>
<td>OSSEC Guide</td>
<td>187</td>
</tr>
<tr>
<td>31.1</td>
<td>Setting Up OSSEC Alerts</td>
<td>187</td>
</tr>
<tr>
<td>31.2</td>
<td>Troubleshooting</td>
<td>190</td>
</tr>
<tr>
<td>31.3</td>
<td>Analyzing the Alerts</td>
<td>194</td>
</tr>
<tr>
<td>32</td>
<td>Setting Up a Printer in Tails</td>
<td>195</td>
</tr>
<tr>
<td>32.1</td>
<td>Installing and Printing via the Tails GUI</td>
<td>196</td>
</tr>
<tr>
<td>32.2</td>
<td>Printing from the Command Line</td>
<td>198</td>
</tr>
<tr>
<td>33</td>
<td>HTTPS on the Source Interface</td>
<td>199</td>
</tr>
<tr>
<td>33.1</td>
<td>Obtaining an HTTPS certificate for Onion URLs</td>
<td>199</td>
</tr>
<tr>
<td>33.2</td>
<td>Activating HTTPS in SecureDrop</td>
<td>201</td>
</tr>
<tr>
<td>34</td>
<td>SSH Over Local Network</td>
<td>203</td>
</tr>
<tr>
<td>34.1</td>
<td>Configuring SSH for Local Access</td>
<td>203</td>
</tr>
<tr>
<td>35</td>
<td>SecureDrop On-Site Training Schedule</td>
<td>205</td>
</tr>
<tr>
<td>35.1</td>
<td>Day 1: Preparation and Install</td>
<td>205</td>
</tr>
<tr>
<td>35.2</td>
<td>Day 2: Journalist and Admin Training</td>
<td>206</td>
</tr>
<tr>
<td>36</td>
<td>Using a YubiKey with the <strong>Journalist Interface</strong></td>
<td>209</td>
</tr>
<tr>
<td>36.1</td>
<td>What is a YubiKey?</td>
<td>209</td>
</tr>
<tr>
<td>36.2</td>
<td>Download and Launch the YubiKey Personalization Tool</td>
<td>209</td>
</tr>
<tr>
<td>36.3</td>
<td>Setting Up Hardware-Based Codes</td>
<td>210</td>
</tr>
<tr>
<td>36.4</td>
<td>Adding Users</td>
<td>211</td>
</tr>
<tr>
<td>36.5</td>
<td>Using Your YubiKey</td>
<td>211</td>
</tr>
<tr>
<td>37</td>
<td>Back Up, Restore, Migrate</td>
<td>213</td>
</tr>
<tr>
<td>37.1</td>
<td>Minimizing Disk Use</td>
<td>213</td>
</tr>
<tr>
<td>37.2</td>
<td>Backing Up</td>
<td>214</td>
</tr>
</tbody>
</table>
SecureDrop is an open-source whistleblower submission system that media organizations can use to securely accept documents from and communicate with anonymous sources.
1.1 Choosing the Right Location

When national security is involved, we suggest you buy a new computer, a USB key and pay cash. In any case you must then find a busy coworking place or cyber cafe you don’t regularly go to and sit at a place with your back to a wall to avoid cameras capturing information on your screen or keystrokes.

1.2 Get the Tor Browser

Each SecureDrop instance has a publicly available Source Interface: a website where sources can create anonymous accounts, submit files and messages, and check back for replies.

Each Source Interface is only available as a Tor Hidden Service, which is a special type of website with an address ending in “.onion” that is only accessible through Tor. Tor is an anonymizing network that makes it difficult for anybody observing the network to associate a user’s identity (e.g. their computer’s IP address) with their activity (e.g. uploading information to SecureDrop).

The easiest and most secure way to use Tor is to download the Tor Browser from the Tor Project website. The Tor Browser is a modified version of the Firefox web browser designed to protect your security and anonymity while using Tor. If there is a chance that downloading the Tor Browser raises suspicion, you have a few alternatives.

- If your mail provider is less likely to be monitored, you can send a mail to gettor@torproject.org and a bot will answer with instructions
- If you routinely use GitHub, you can use it to download the Tor Browser

Once you have the Tor Browser, launch it and enter the “.onion” address for the Source Interface of the organization that you wish to submit to. You can find this address on the organization’s Landing Page, or listed on the SecureDrop Directory.

While using the Tor Browser on your personal computer helps hide your activity on the network, it leaves traces (of its own installation) on your local machine. For even more deniability, we recommend booting into a live system such as Tails for a higher level of security. Tails is specifically designed to run on your computer without leaving traces of...
your activity, and automatically routes all of your Internet browsing through Tor so you can easily access SecureDrop safely.

### 1.3 Choose Who to Submit To

Each SecureDrop instance is totally independent, and submissions to that instance are only available to journalists associated with that organization.

All organizations have a Landing Page that provides their own organization-specific recommendations for using SecureDrop. We encourage you to consider an organization’s Landing Page before submitting to them.

Most organizations make their Landing Page prominently accessible from their main website’s homepage. You can also find an incomplete list of organizations accepting submissions through SecureDrop on the SecureDrop Directory maintained by Freedom of the Press Foundation.

### 1.4 Making Your First Submission

Open the Tor Browser and navigate to the .onion address for the SecureDrop Source Interface you wish to make a submission to. The page should look similar to the screenshot below, although it will probably have a logo specific to the organization you are submitting to.

If this is the first time you’re using the Tor Browser, it’s likely that you have JavaScript enabled and that the Security Setting that the Tor Browser provides is set to “Low”. If you do, there will be a purple warning banner at the top of the page that encourages you to disable JavaScript and turn up the Security Setting to Safest:

Click the Security Setting link in the warning banner and a message bubble will pop up explaining how to disable JavaScript and configure your security properly:

Follow the instructions and the page should refresh automatically. Note that this will change your security settings and disable JavaScript for every page in your Tor Browser, and this setting will persist across browser sessions.
The page should look similar to the screenshot below. If this is the first time you are using SecureDrop, click the **Get Started** button.

You should now see a screen that shows the unique codename that SecureDrop has generated for you. Note that your codename will not be the same as the codename shown in the image below. It is extremely important that you both remember this code and keep it secret. After submitting documents, you will need to provide this code to log back in and check for responses.

The best way to protect your codename is to memorize it. If you cannot memorize it right away, we recommend writing it down and keeping it in a safe place at first, and gradually working to memorize it over time. Once you have memorized it, you should destroy the written copy.

**Tip:** For detailed recommendations on best practices for managing your passphrase, check out *Passphrase Best*.
Once you have generated a codename and put it somewhere safe, click **Submit Documents**.

You will next be brought to the submission interface, where you may upload a document, enter a message to send to journalists, or both. You can only submit one document at a time, so you may want to combine several files into a zip archive if necessary. The maximum submission size is currently 500MB. If the files you wish to upload are over that limit, we recommend that you send a message to the journalist explaining this, so that they can set up another method for transferring the documents.

When your submission is ready, click **Submit**.
After clicking **Submit**, a confirmation page should appear, showing that your message and/or documents have been sent successfully. On this page you can make another submission or view responses to your previous messages.
Once you are finished submitting documents, be certain you have saved your secret codename and then click the **Log out** button:

The final step to clearing your session is to restart Tor Browser for optimal security. You can either close the browser entirely or follow the notification: click on the Tor onion in the toolbar, click **New Identity** and then click **Yes** in the dialog box that appears to confirm you’d like to restart Tor Browser:
1.5 Continuing the Conversation

If you have already submitted a document and would like to check for responses, click the Log in button on the media organization’s SecureDrop homepage.

The next page will ask for your secret codename. Enter it and click Continue.

If a journalist has responded, their message will appear on the next page. This page also allows you to upload another document or send another message to the journalist. Before leaving the page, you should delete any replies. In the unlikely event that someone learns your codename, this will keep your identity secret as no one will be able to see the previous correspondences you had with journalists.
After you delete the message from the journalist, make sure you see the below message.
SecureDrop Documentation, Release 1.0.0

If the server experiences a large number of new sources signing up at once and is overloaded with submissions, the journalist will flag your message on their end and you will see the message below. They can’t write a reply to you until you’ve seen this message for security reasons. This will only happen the first time a journalist replies and with subsequent replies you will skip this step. Click Refresh or log in again to see if a journalist has responded.
Repeat these steps to continue communicating with the journalist.
This guide presents an overview of the SecureDrop system for a journalist. It covers the core functions necessary to start working with the platform: logging in securely, viewing documents, editing documents, and interacting with sources.

2.1 Updating Your Workstation

You should keep your SecureDrop workstations updated with:

- Tails updates
- SecureDrop workstation updates

You should apply Tails updates to your Tails drive as they are released, as they often contain critical security fixes. The Journalist Workstation Tails drive, once booted and connected to Tor, will alert you if upgrades are available. For most Tails upgrades, you can simply follow the steps in the Tails Upgrader that appears on screen to update your Tails drive. However, sometimes Tails upgrades are “manual” which means that you should follow the instructions in Tails Upgrade Documentation to upgrade the drives. Talk to your SecureDrop administrator if you have trouble.

For SecureDrop workstation updates, beginning with SecureDrop 0.7.0, your workstation will automatically check for updates on boot. An update window will pop up when updates are needed, and you should simply follow the prompts in the updater to perform the update.

Note: Note that you will need to have a Tails Administrator password configured to complete the update. If you forget to do so, you will need to reboot to enable it.
2.2 Connecting to the **Journalist Interface**

Journalists viewing documents on SecureDrop must connect to the *Journalist Interface* using the Tails operating system on a USB drive. As part of your on-boarding, your admin should have provided you with a Tails drive configured for this purpose, known as the *Journalist Workstation* USB drive.

If you do not have a USB drive clearly identified as the *Journalist Workstation*, ask your administrator for assistance before continuing.

**Note:** The Tails OS makes using SecureDrop very different from other computing experiences. The added layers of security mean extra steps each time you want to login. With practice, you will become increasingly comfortable with the process.

To use the *Journalist Interface*, you will visit a Tor Onion Service address in the Tor Browser. By design, this Onion Service address is only accessible from your *Journalist Workstation*; it will not work in Tor Browser on another computer, unless explicitly configured with an access token.

To visit the *Journalist Interface*, click the *Journalist Interface* icon on the desktop. This will open Tor Browser to a “.onion” address. Log in with your username, passphrase, and two-factor authentication token, as shown in the first screenshot below. (If you have been provided with a YubiKey, see *Using YubiKey with the Journalist Interface* for detailed setup and usage information.)

![Login to access the journalist interface](https://securedropproject.org/images/login_to_access_journalist_interface.png)

### 2.2.1 Reset Passphrase or Two-factor Authentication Token

If necessary, journalists may reset their user passphrase or two-factor authentication token in their user profile. To navigate to your user profile, log in to the *Journalist Interface* and click on the link in the upper right of the screen where it says **Logged on as <your user name>**.

If you have lost or forgotten your passphrase or two-factor authentication device, you will need to contact your SecureDrop admin for assistance.
2.3 Daily Journalist Alerts About Submissions

When a SecureDrop has little activity and receives only a few submissions every other week, checking the Journalist Interface daily only to find there is nothing is a burden. It is more convenient for journalists to be notified daily via encrypted email about whether or not there has been submission activity in the past 24 hours.

If the email shows submissions were received, the journalist can connect to the Journalist Interface to get them.

This is an optional feature that must be activated by the administrator. In the simplest case a journalist provides their email and GPG public key to the admin. If a team of journalist wants to receive these daily alerts, they should share a GPG key and ask the admin to setup a mail alias (SecureDrop does not provide that service) so they all receive the alerts and are able to decrypt them.
2.4 Interacting With Sources

If any sources have uploaded documents or sent messages, they will be listed on the homepage by codename.

Note: Codenames that journalists see are different than the codenames visible to sources.

Click on a codename to see the dedicated page for that source. You will see all of the messages that they have written and documents that they have uploaded. If the name of a source is difficult to say or remember, you can rename a source using the Change codename link next to their current codename.

Tip: You can also Star interesting or promising sources to easily return to them later. All starred sources will be bumped to the top of the list of sources.

If you want to reply to the source, write your message in the text field and click Submit.

Once your reply has been successfully submitted, you will be returned to the source page and see a message confirming that the reply was stored. The source will see your reply the next time they log in with their unique codename.

To minimize the impact of a source codename being compromised, the Source Interface encourages the source to delete the reply after reading it. Once a source has read your reply and deleted it from their inbox, a checkmark will appear next to the reply in the interface.

Note: Prior to SecureDrop 0.9.0, replies when deleted from the source inbox would also disappear from the journalist inbox. As such, if there are older conversations, there may be discontinuities in the conversation.
You may also delete replies if you change your mind after sending them.

Documents and messages are encrypted to the SecureDrop installation’s Submission Public Key. In order to read the messages or look at the documents you will need to transfer them to the Secure Viewing Station, which holds the Submission Private Key. To recall the conversation history between your organization and sources, you can also download replies and transfer them to the Secure Viewing Station for decryption.

### 2.4.1 Flag for Reply

If the server experiences a large number of new sources signing up at once and is overloaded with submissions, you will need to flag sources for reply before you can communicate with them. Click the **Flag this source for reply** button.

After clicking the **Flag this source for reply** button, you will see this confirmation page. Click through to get back to the page that displays that source’s documents and replies.
You will not be able to reply until after the source logs in again and sees that you would like to talk to them. So you may have to sit and wait. After the source sees that you’d like to reply, a GPG key pair will automatically be generated and you can log back in and send a reply.

### 2.5 Moving Documents to the Secure Viewing Station

#### 2.5.1 Step 1: Download the encrypted submission

Documents and messages sent by sources can only be decrypted and viewed on the Secure Viewing Station. After clicking on an individual source, you will see the page below with any documents or messages the source has sent you. Documents always end with `-doc.gz.gpg`, while messages always end with `-msg.gpg`.

Click on a document or message name to save it, or select a number of documents and save them all at once by clicking Download Selected.

A dialog box with two choices will appear, Cancel and Save file:
Click **Save file**. In the save dialog, select one of the two folders highlighted in red in the screenshot below:

The difference between these two folders is as follows:

- **Tor Browser**: Downloads saved to this folder will be stored in memory, which means that they will only be available for the duration of your current Tails session. In the screenshot, this is the currently selected folder. The full path to this folder is `/home/amnesia/Tor Browser`.

- **Tor Browser (persistent)**: Note that the name may be abbreviated, as shown in the screenshot; you can view the full name by hovering the mouse over the shortcut. Downloads saved to this folder will be stored on your Tails USB drive in the special persistent volume that is only available if you have unlocked it on the Tails welcome screen. The full path to this folder is `/home/amnesia/Persistent/Tor Browser`.

Unless you have a reason to store encrypted submissions on the *Journalist Workstation*, we recommend using the non-persistent “Tor Browser” folder. In the recommended process, you will now move the submission to the *Secure Viewing Station*, and there is no need to leave a persistent copy behind.

**Important:** Attempting to download files to any other folder will fail. Tails only permits Tor Browser to access these two folders, so that even if your browser is compromised by malware, attackers cannot easily gain access to other data stored on the same computer.

See the Tails guide to *Browsing the web with Tor Browser* for more information.

### 2.5.2 Step 2: Copy the encrypted submission to the *Transfer Device*

Once downloaded to either folder, move the document to the designated USB stick you intend to use to transfer the documents from your *Journalist Workstation* to the *Secure Viewing Station*. This storage device is known as your *Transfer Device*.

**Note:** If the *Transfer Device* was set up according to our recommendations, you will be prompted for a decryption passphrase on the *Journalist Workstation* and the *Secure Viewing Station* before being able to use it in a given session. We recommend storing this passphrase in your own personal password manager (e.g., on your smartphone), so that it is readily accessible to you whenever you need it.

You can right-click the file and select **Copy to**, then select the *Transfer Device*, as shown in the screenshots below.
This will leave a redundant copy behind in the Tor Browser folder. If you have downloaded the file to the non-persistent “Tor Browser” folder (as recommended), the redundant copy will disappear when the computer is shut down or rebooted.

“Eject” the Transfer Device by clicking the eject icon next to its name in the file manager. Wait for this operation to complete (the eject icon will disappear), then unplug the Transfer Device. “Ejecting” the drive in this manner ensures that all write operations are completed before you physically unplug it.

2.5.3 Step 3: Decrypt and view the submission on the Secure Viewing Station

Next, boot up the Secure Viewing Station using Tails and enter the passphrase for the Secure Viewing Station persistent volume. Once you have logged in, plug in the Transfer Device.

Note: The Secure Viewing Station and Journalist Workstation are on separate Tails USB drives.

Click on the Home icon on your desktop, then on the Transfer Device. Copy the file into your Persistent folder. You can do so by opening a new window with the Persistent folder and dragging the file from one window to another. A faster method is to drag the file to the Persistent shortcut, as in the screenshot below:
Important: Always copy submissions to the Persistent folder before decrypting them. Otherwise you might accidentally decrypt them on the USB stick, and they could be recoverable in the future.

After successfully copying them to the Secure Viewing Station, erase the files from your Transfer Device. Ensure you’re viewing the Transfer Device folder, then right click on the files that need removal and click “Wipe” to securely delete the files from your device:

To decrypt and view documents or messages, return to your Persistent folder. All key actions are initiated by double-clicking:

- Double-clicking archives in ZIP or gzip format will open the “File Roller” application, which allows you to extract the contents.
- Double-clicking files that end in .gpg will attempt to decrypt the contents to the same directory. If you have configured a passphrase for your Submission Key, you will be prompted for it.
- Double-clicking decrypted messages or documents will attempt to open them in a default application suitable for the file type.

If the default application does not work, you can right-click on the document and choose Open with Other Application... to try opening the document with LibreOffice Writer, Document Viewer, or another application. You might also need to right-click on a file and choose Rename... to rename a document with an incorrect or missing file extension.

Tip: Always extract gzip archives with the “File Roller” application, which is the default when double-clicking the archive. Other methods may not preserve the filename contained in the archive.
For example, an archive called 1-artful_elevation-doc.gz might contain a file secrets.docx, but if you extract the contents by right-clicking the archive and selecting Extract here, the extracted file will be called 1-artful_elevation-doc instead of secrets.docx. This may result in problems when attempting to open the file due to the loss of its file extension.

When you double-click an archive to open it, you should see it in the “File Roller” application. It looks like this:

Click the Extract button to unpack the archive. Navigate to the folder containing the encrypted document message or document (ends with .gpg) and double-click it to decrypt it. The decrypted file it will have the same filename, but without .gpg at the end.

You can now double-click on the decrypted file to open it in its default application.
2.6 Working with Documents

This section describes how to organize submissions, handle unusual file formats, safely research submissions, remove metadata, and mitigate risks from submitted malware.

Tip: This is only a very limited introduction. Freedom of the Press Foundation publishes and maintains digital security guides for journalists, many of which relate to these topics, and offers digital security training for news organization staff.

2.6.1 Organizing submissions

Whenever you download submissions using one of the Download buttons in the Journalist Interface, they will be organized as a ZIP archive with a built-in folder structure, which you can use as a template for organizing submissions on the Secure Viewing Station.

Submissions downloaded in this manner from the list of all sources will contain a structure like the following:

```
all
  recessive accreditation
    1_2019-07-07
      1-recessive_accreditation-msg.gpg
    2_2019-07-07
      2-recessive_accreditation-msg.gpg
  surviving authentication
    1_2019-07-07
      1-surviving_authentication-doc.gz.gpg
    2_2019-07-07
      2-surviving_authentication-msg.gpg
```

Submissions downloaded in this manner from the screen for an individual source will contain a similar structure, but without the parent folder all.

A folder like 1_2019-07-07 in the example above will always contain exactly one message or document. The numbers in the folder name (1, 2, etc.) correspond to the numbering in the Journalist Interface. The dates (2019-07-07 in the example above) are the day (in year/month/day format) of the last activity related to this source, at the time the archive was downloaded.

If you download messages or documents one at a time in the Journalist Interface, they will not be contained in a ZIP file at all. Instead, you will be dealing with individual files like 1-surviving_authentication-doc.gz.gpg, without a folder structure.

2.6.2 Handling File Formats

SecureDrop accepts submissions of any file type. Tails comes with pre-installed applications for securely working with documents, including an office suite, graphics tools, desktop publishing tools, audio tools, and printing and scanning tools.

For more information, visit the Tails guide to working with sensitive documents.

2.6.3 Pre-Encrypted Submissions

SecureDrop sources can optionally encrypt prior to submitting to SecureDrop. This means that once you decrypt the document as you usually do by double clicking the document in the file manager, there will be another layer of
Most often, the file will be encrypted to the SecureDrop key. If the file is encrypted to your SecureDrop key, you should be able to double click the file as usual once more in the SVS and it should decrypt.

However, it’s also possible the file is encrypted to another key, potentially your personal key. If this occurs, you will get an error message in Tails that reads “Decryption failed. You probably do not have the decryption key”. To determine which key was used, if you are comfortable at the command line, you can open the Terminal, navigate to the file, and use:

```sh
gpg --decrypt NAME_OF_FILE
```

replacing `NAME_OF_FILE` with the name of the file you wish to decrypt. This command will tell you what key was used to encrypt the file. If you are not comfortable at the command line, contact your SecureDrop admin or Freedom of the Press Foundation for assistance.

**Warning**: Do not transfer source material off the Secure Viewing Station for decryption. Instead, transfer cryptographic keys to the SVS device for decryption and metadata removal.

### 2.6.4 Researching Submissions

Journalists should take care to research submissions using the Tor Browser, ideally in a new Tails session on your Journalist Workstation for highly sensitive submissions.

### 2.6.5 Removing Metadata

**Tip**: For detailed information about removing metadata from documents, check out this in-depth guide to removing metadata.

Tails comes with the Metadata Anonymisation Toolkit (MAT) that is used to help strip metadata from a variety of types of files, including png, jpg, OpenOffice/LibreOffice documents, Microsoft Office documents, pdf, tar, tar.bz2, tar.gz, zip, mp3, mp2, mp1, mpa, ogg, and flac. You can open MAT by clicking Applications System Tools MAT.

We recommend always doing as much work as possible inside of Tails before working with documents on your everyday workstation. This includes stripping metadata with MAT.

**Warning**: MAT is no longer actively maintained and will not strip all metadata, even when the output claims the document is clean. Some metadata are likely to persist: you must never assume MAT has removed all metadata.

When you no longer need documents, you can right-click on them and choose **Wipe** to delete them.
2.6.6 Risks From Malware

SecureDrop does not scan for or remove malware in submissions you receive. There are important steps you can take to protect yourself:

1. **Keep the version of Tails on your Secure Viewing Station up-to-date.**
   
   Tails offers more protection against compromise than your everyday computer, and the air-gap prevents potential malware from “phoning home.” But if the version of Tails is outdated, an attacker can still attempt to exfiltrate or destroy information.

2. **Print documents from the Secure Viewing Station instead of exporting them digitally, whenever possible.**

   Printing documents prevents the proliferation of malware to your everyday workstation, and eliminates most categories of embedded metadata. Note that printing a document may still preserve watermarks, printer codes, steganographically encoded data, or other information not visible to the naked eye.

3. **Consult with your administrator or your digital security staff before copying files digitally.**

   If you must copy a file in digital form (because of its format, the volume of information, or for other reasons), we recommend taking the time to consult with technical experts within the organization.

   **Tip:** Converting files to simpler formats (e.g., PDF to PNG) can help reduce the risk of malware. Tails provides both graphical and command-line utilities that can be used for this purpose.

4. **Never scan QR codes embedded in documents using a network-connected device.**

   QR codes can contain malicious links that your device will automatically visit. In general, be careful when opening any links provided in a SecureDrop submission, as this can leak information to third parties. If you are unsure if a link is safe to click, you should consult internally, or contact Freedom of the Press Foundation for assistance.

5. **Don’t photograph submissions using your smartphone, and be careful with all digital photography.**

   Many smartphones are configured to back up photographs to cloud services, immediately or intermittently; newer digital cameras have similar functionality. Not all backup settings may be visible to you.

   Any digital photograph will include certain metadata by default, which may reveal sensitive information about your SecureDrop usage patterns (potentially including GPS coordinates) to anyone who gains access to the file.

Fully mitigating the risks of malware received via SecureDrop is beyond the scope of this documentation. If you have questions, you can contact us at securedrop@freedom.press (GPG encrypted) or via the support portal. Please do NOT disclose details about the contents of any submission you have received.
2.7 Moving Documents to Your Everyday Workstation

**Important:** As noted above, SecureDrop does not scan for or remove malware. If the file you received contains malware targeting the operating system and applications running on your everyday workstation, copying it in its original form carries the risk of spreading malware to that computer. Make sure you understand the risks, and consider other methods to export the document (e.g., print).

If you must copy a file from your *Secure Viewing Station* to your everyday workstation in digital form, our *recommendation* is that journalists are provided with an *Export Device*, typically a USB drive, which is encrypted using *VeraCrypt*. These instructions assume that you are following the recommended workflow. If you are unsure, ask your administrator.

**Note:** VeraCrypt support was added to Tails in version 3.9. We strongly recommend keeping your *Secure Viewing Station* up-to-date with each release of Tails.

To open the *Export Device* on the *Secure Viewing Station*, follow these steps:

1. If your *Export Device* has a physical write protection switch, make sure it is in the *unlocked* position.
2. Plug the *Export Device* into the *Secure Viewing Station*.
3. Click *Applications Utilities Unlock VeraCrypt Volumes*
4. Under “Partitions and Drives”, select the *Export Device* and click *Unlock*.
5. Enter your passphrase, which we recommend keeping in your own personal password manager (e.g., on your smartphone), not on KeePassX.
6. Under “Partitions and Drives”, open the encrypted drive by clicking *Open*.

Steps 3-6 are illustrated below:
The Export Device should now open in the file manager. If there are still files on the Export Device from your last copy operation, securely delete them now.

Copy the file or files you want to access on your everyday workstation to the Export Device using the file manager.

### 2.8 Decrypting and Preparing to Publish
Note: To decrypt a VeraCrypt drive on a Windows or Mac workstation, you need to have the VeraCrypt software installed. If you are unsure if you have the software installed or how to use it, ask your administrator, or see the Freedom of the Press Foundation guide for working with VeraCrypt.

To access the Export Device on your everyday workstation, follow these steps:

1. If your Export Device has a physical write protection switch, make sure it is in the locked position.
2. Plug the Export Device into your everyday workstation.
3. Launch the VeraCrypt application.
4. Click Select Device and select the Export Device, then click OK.
5. Click Mount.
6. Enter the passphrase for your Export Device. You should find this in your own personal password manager.
7. Open the Export Device in your operating system’s file manager, and copy the contents of interest to your everyday workstation.

As a security precaution, we recommend deleting the files on the Export Device after each copy operation. If you are using write protection, you have to perform this step on the Secure Viewing Station to get the security benefits of write protection.

When you are done, switch back to the VeraCrypt window, and click Dismount.

You are now ready to write articles and blog posts, edit video and audio, and begin publishing important, high-impact work!

Tip: Check out our SecureDrop Promotion Guide to read about encouraging sources to use SecureDrop.
3.1 Responsibilities

The SecureDrop architecture contains multiple machines and hardened servers. While we have automated many of the installation and maintenance tasks, a skilled Linux admin is required to responsibly run the system.

This section outlines the tasks the admin is responsible for in order to ensure that the SecureDrop server continues to be a safe place for sources to talk to journalists.

3.1.1 Maintaining Credentials

The admin should have her own username, passphrase, and two-factor authentication method (via smartphone app or YubiKey). Admins are also responsible for managing user credentials and encouraging best practices. (See Passphrases and Passphrase Best Practices.)

3.1.2 Updating the SecureDrop Servers

The admin should be aware of all SecureDrop updates and take any required manual action if requested in the SecureDrop Release Blog. We recommend subscribing to the SecureDrop RSS Feed to stay apprised of new updates.

Most often, the SecureDrop server will automatically update via apt. However, occasionally you will need to run securedrop-admin install. We will inform you in the release blog when this is the case. If you are onboarded to our SecureDrop Support Portal, we will let you know in advance of major releases if manual intervention will be required.
3.1.3 Updating the Network Firewall

Given all traffic first hits the network firewall as it faces the non-Tor public network, the admin should ensure that critical security patches are applied to the firewall.

Be informed of potential updates to your network firewall. If you’re using the network firewall suggested by FPF, you can subscribe to the Netgate RSS Feed to be alerted when releases occur. If critical security updates need to be applied, you can do so through the firewall’s pfSense WebGUI. Refer to our Keeping pfSense up to Date documentation or the official pfSense Upgrade Docs for further details on how to update the suggested firewall.

3.1.4 Updating the SecureDrop Workstations

The admin should keep all SecureDrop workstations updated with

- **Tails updates** for each Admin Workstation, Journalist Workstation, and Secure Viewing Station; and
- **SecureDrop workstation updates** for each Admin Workstation and Journalist Workstation.

You should apply Tails updates to your Tails drives as they are released, as they often contain critical security fixes. Subscribe to the Tails RSS Feed to be alerted of new releases. The online Tails drives, once booted and connected to Tor, will alert you if upgrades are available. Follow the Tails Upgrade Documentation on how to upgrade the drives.

For SecureDrop workstation updates, beginning with SecureDrop 0.7.0, your workstation will automatically check for updates on boot. An update window will pop up when updates are needed, and you should simply follow the prompts in the updater to perform the update.

**Note:** Note that you will need to have a Tails Administrator password configured to complete the update. If you forget to do so, you will need to reboot to enable it.

3.1.5 Monitoring OSSEC Alerts for Unusual Activity

The admin should decrypt and read all OSSEC alerts. Report any suspicious events to FPF through the SecureDrop Support Portal. See the OSSEC Guide for more information on common OSSEC alerts.

**Warning:** Do not post logs or alerts to public forums without first carefully examining and redacting any sensitive information.

**Note:** You can send a test OSSEC alert to verify OSSEC and your email configuration is working properly through the Admin Interface by clicking Send Test OSSEC Alert:
3.2 Common Tasks

3.2.1 Adding Users

Now you can add new logins for the journalists at your news organization who will be checking the system for submissions. Make sure the journalist is physically in the same room as you when you do this, as they will have to scan a barcode for their two-factor authentication. Since you’re logged in, this is the screen you should see now:

In the top right corner click the “Admin” link, which should bring you to this page:
Once there, click ‘Add User’ button, which will take you to this page:

Here, you will hand the keyboard over to the journalist so they can create their own username. Once they’re done entering a username for themselves, have them write down their pre-generated diceware passphrase. Then, you will select whether you would like them to also be an admin (this allows them to add or delete other journalist accounts), and whether they will be using FreeOTP or a YubiKey for two-factor authentication.

**FreeOTP**

If they are using FreeOTP for their two-factor, they can just proceed to the next page:
At this point, the journalist should make sure they have downloaded the FreeOTP app to their smartphone. It can be installed from the Apple Store for an iPhone or from the Google Play store for an Android phone. Once you download it and open it, the app does not require setup. It should prompt you to scan a barcode. The journalist should use their phone’s camera to scan the barcode on the screen.

If they have difficulty scanning the barcode, they can tap on the icon at the top that shows a plus and the symbol of a key and use their phone’s keyboard to input the random characters that are highlighted in yellow, in the Secret input field, without white space.

Inside the FreeOTP app, a new entry for this account will appear on the main screen, with a six digit number that recycles to a new number every thirty seconds. Enter the six digit number under “Verification code” at the bottom of the screen, and hit enter.

If FreeOTP was set up correctly, you will be redirected back to the Admin Interface and will see a confirmation that the two-factor token was verified.

**Tip:** We recommend using FreeOTP (available for Android and for iOS) to generate two-factor authentication tokens because it is Free Software. However, if it does not work for you for any reason, alternatives exist:

- Google Authenticator for Android and iOS (proprietary)
- authenticator for the desktop (Free Software)
YubiKey

If the journalist wishes to use a YubiKey for two-factor authentication, check the box next to “I’m using a YubiKey”. You will then need to enter the OATH-HOTP Secret Key that your YubiKey is configured with. For more information, read the YubiKey Setup Guide.

Once you’ve configured your YubiKey and entered the Secret Key, click Add user. On the next page, enter a code from your YubiKey by inserting it into the workstation and pressing the button.

If everything was set up correctly, you will be redirected back to the Admin Interface, where you should see a flashed message that says “Two factor token successfully verified for user new username!”.

Congratulations! You have successfully set up a journalist on SecureDrop. Make sure the journalist remembers their username and passphrase and always has their 2 factor authentication device in their possession when they attempt to log in to SecureDrop.

3.2.2 Server Command Line Use

Generally, you should avoid directly SSHing into the servers in favor of using the Admin Interface or securedrop-admin CLI tool. However, in some cases, you may need to SSH in order to troubleshoot and fix a problem that cannot be resolved via these tools.

In this section we cover basic commands you may find useful when you SSH into the Application Server and Monitor Server.

Tip: When you SSH into either SecureDrop server, you will be dropped into a tmux session. tmux is a screen multiplexer - it allows you to tile panes, preserve sessions to keep your session alive if the network connection fails, and more. Check out this tmux tutorial to learn how to use tmux.

Tip: If you want a refresher of the Linux command line, we recommend this resource to cover the fundamentals.
Shutting Down the Servers

```
sudo shutdown now -h
```

Rebooting the Servers

```
sudo reboot
```

Investigating Logs

Consult our Investigating Logs topic guide for locations of the most relevant log files you may want to examine as part of troubleshooting, and for how to enable error logging for the Source Interface.

**Note:** You can use the securedrop-admin tool to extract logs to send to Freedom of the Press Foundation for analysis. Run the following command on your Admin Workstation:

```
cd ~/Persistent/securedrop
./securedrop-admin logs
```

This command will produce encrypted tarballs containing logs from each server. See the command output for more information.

Immediately Apply a SecureDrop Update

SecureDrop will update and reboot once per day. However, if after a SecureDrop update is announced you wish to fetch the update immediately, you can SSH into each server and run:

```
sudo cron-apt -i -s
```

Application Server

Adding Users (CLI)

After the provisioning of the first admin account, we recommend using the Admin Interface web application for adding additional journalists and admins.

However, you can also add users via `./manage.py` in `/var/www/securedrop/` as described during first install. You can use this command line method if the web application is unavailable.

Restart the Web Server

If you make changes to your Apache configuration, you may want to restart the web server to apply the changes:

```
sudo service apache2 restart
```
Cleaning up deleted submissions

When submissions are deleted through the web interface, their database records are deleted and their encrypted files are securely wiped. For large files, secure removal can take some time, and it's possible, though unlikely, that it can be interrupted, for example by a server reboot. In older versions of SecureDrop this could leave a submission file present without a database record.

As of SecureDrop 1.0.0, automated checks send OSSEC alerts when this situation is detected, recommending you run `manage.py list-disconnected-fs-submissions` to see the files affected. As with any `manage.py` usage, you would run the following on the admin workstation:

```
ssh app
sudo -u www-data bash
cd /var/www/securedrop
./manage.py list-disconnected-fs-submissions
```

You then have the option of running:

```
./manage.py delete-disconnected-fs-submissions
```

to clean them up. As with any potentially destructive operation, it's recommended that you *back the system up* before doing so.

There is also the inverse scenario, where a database record could point to a file that no longer exists. This would usually only have happened as a result of disaster recovery, where perhaps the database was recovered from a failed hard drive, but some submissions could not be. The OSSEC alert in this case would recommend running:

```
./manage.py list-disconnected-db-submissions
```

To clean up the affected records you would run (again, preferably after a backup):

```
./manage.py delete-disconnected-db-submissions
```

Even when submissions are completely removed from the application server, their encrypted files may still exist in backups. We recommend that you delete old backup files with `shred`, which is available on Tails.

Monitor Server

Restart OSSEC

If you make changes to your OSSEC monitoring configuration, you will want to restart OSSEC via OSSEC’s control script, `ossec-control`:

```
sudo /var/ossec/bin/ossec-control restart
```

Updating the Servers

Sometimes you will want to update the system configuration on the SecureDrop servers. For example, to customize the logo on the source interface, or change the PGP key that OSSEC alerts are encrypted to. You can do this from your *Admin Workstation* by following the procedure described in this section.
3.2.3 Updating Logo Image

You can update the system logo shown on the web interfaces of your SecureDrop instance via the Admin Interface. We recommend a size of 500px × 450px. Simply click the **Update Instance Config** button:

![Update Instance Config Button](image)

And on the instance configuration page, select and upload the PNG image you prefer. You should see a message appear indicating the change was a success:
3.2.4 Updating System Configuration

If you want to update the system configuration, you should use the securedrop-admin tool on the Admin Workstation. From ~/Persistent/securedrop, run:

```
./securedrop-admin sdconfig
```

This will give you the opportunity to edit any variable. Answer the prompts with values that match your environment. An example of one such prompt would be to set the daily reboot time. To minimize the presence/duration of plaintext in memory, the servers are rebooted every 24 hours to periodically wipe the memory. As an admin, you can configure this automatic reboot time. By default, it is set at 4:00 a.m. and you can change it to suit your timing. Next, you will need to apply the changes to the servers. Again from ~/Persistent/securedrop:

```
./securedrop-admin install
```

Note: If you see an error running ./securedrop-admin install, and believe it may be an intermittent issue (for example, due to losing network connectivity to the servers), it is safe to run the ./securedrop-admin install command again. If you see the same issue consistently, then you will need to troubleshoot it.
If you see the error message “timeout (62s) waiting for privilege escalation prompt”, try deleting the Ansible control path directory on your *Admin Workstation* (`rm -rf ~/.ansible/cp`) to reset the connection to the servers, then re-run the `./securedrop-admin install` command from within `~/Persistent/securedrop`.

If you encounter other errors, we encourage you to submit a bug report, or to contact us at securedrop@freedom.press (GPG encrypted).

Once the install command has successfully completed, the changes are applied. Read the next section if you have multiple admins.

---

**Note:** Server configuration is stored on the *Admin Workstation* in `~/Persistent/securedrop/install_files/ansible-base/group_vars/all/site-specific`.

---

### 3.2.5 Managing site-specific Updates On Teams With Multiple Admins

Organizations with multiple admins should establish a protocol to communicate any changes one admin makes to the site-specific configuration file on the server.

Currently, when one admin pushes changes in `site-specific` to the server, the changes will not sync to the local `site-specific` file on the remaining admin workstations. Without being aware of changes made to `site-specific`, admins run the risk of pushing old information to the servers. This can affect the receipt of OSSEC alerts, viability of the Submission Key, among other critical components of the SecureDrop environment.

There are multiple ways to avoid pushing out-of-date information to the servers. We recommend admins establish a secure communication pipeline to alert fellow admins of any changes made to `site-specific` on the server. That clues every admin in on changes in real time, providing all team members with a reminder to manually update all `site-specific` files.

In addition to secure group communications, admins can learn of updates to the server by monitoring OSSEC alerts. (Please note that while an OSSEC alert can notify you of the occurrence of an update to the server, it may not reveal the content of the change.) Another management option would be SSHing into the server and manually inspecting the configuration to identify any discrepancies.

### 3.2.6 Configuring Localization for the Source Interface and the Journalist Interface

The *Source Interface* and *Journalist Interface* are translated in the following languages:

- Arabic (`ar`)
- Catalan (`ca`)
- German (`de_DE`)
- Greek (`el`)
- Spanish (`es_ES`)
- French (`fr_FR`)
- Hindi (`hi`)
- Icelandic (`is`)
- Italian (`it_IT`)
- Norwegian (`nb_NO`)
- Dutch (`nl`)
At any time during and after initial setup, you can choose from a list of supported languages to display using the codes shown in parentheses.

**Note:** With a Source Interface displayed in French (for example), sources submitting documents are likely to expect a journalist fluent in French to be available to read the documents and follow up in that language.

To add or remove locales from your instance, you’ll need to update your system configuration as outlined above.

When you reach the prompt starting with “Space separated list of additional locales to support”, you will see a list of languages currently supported. Refer to the list above to see which languages correspond to which language codes. For example:

```
Space separated list of additional locales to support (ru nl pt_BR fr_FR tr it_IT zh_Hant sv hi ar en_US de_DE es_ES nb_NO): nl fr_FR es_ES
```

You’ll need to list all languages you now want to support, adding or removing languages as needed. Locale changes will be applied after the next reboot.

### 3.3 Frequently Asked Questions

Some initial troubleshooting steps for common scenarios follow. If you continue to have trouble after following these steps, you can contact the SecureDrop team for further assistance.

#### 3.3.1 Generic Troubleshooting Tips

When troubleshooting, ensure you are on the latest version of SecureDrop in your Admin Workstation. This is done by accepting the update when prompted at boot in the GUI that appears.

#### 3.3.2 I can’t SSH into my servers over Tor from my Admin Workstation. What do I do?

At any point after the successful installation of SecureDrop, if you cannot SSH into your Admin Workstation, you should first perform the following troubleshooting steps:

1. **Ensure that you are connected to Tor.** You can do this by browsing to any site in Tor Browser in your Admin Workstation.

2. **Ensure your servers are online.** Visit the Admin Interface to check your Application Server is online, and you can trigger a test OSSEC alert to verify your Monitor Server is online.

3. **Ensure that SSH aliases and onion service authentication are configured:**
• First, ensure that the correct configuration files are present in ~/Persistent/securedrop/install_files/ansible-base.

If v2 onion services are configured, you should have 4 files:
- app-ssh-aths
- mon-ssh-aths
- app-journalist-aths
- app-source-ths

If v3 onion services are enabled, you should have the following 5 files:
- app-ssh.auth_private
- mon-ssh.auth_private
- app-journalist.auth_private
- app-sourcev3-ths
- tor_v3_keys.json

• Then, from ~/Persistent/securedrop, run ./securedrop-admin tailsconfig. This will ensure your local Tails environment is configured properly.

4. **Confirm that your SSH key is available**: During the install, you configured SSH public key authentication using ssh-copy-id. Ensure this key is available using ssh-add -L. If you see the output “This agent has no identities.” then you need to add the key via ssh-add prior to SSHing into the servers.

3.3.3 I got a unusual error when running ./securedrop-admin install. What do I do?

If the error message is not informative, try running it again. The Tor connection can be flaky and can cause apparent errors, but there is no negative impact of re-running ./securedrop-admin install more than once. The command will simply check which tasks have been completed, and pick up where it left off. However, if the same issue persists, you will need to investigate further.
All SecureDrop users—Sources, Journalists, and Admins—are required to memorize at least one passphrase. This document describes best practices for passphrase management in the context of SecureDrop.

### 4.1 General Best Practices

1. **Do** memorize your passphrase.
2. If necessary, **do** write your passphrase down temporarily while you memorize it.

   **Caution:** **Do** store your written passphrase in a safe place, such as a safe at home or on a piece of paper in your wallet. **Do** destroy the paper as soon as you feel comfortable that you have the passphrase memorized. **Do not** store your passphrase on any digital device, such as your computer or mobile phone.

3. **Do** review your passphrase regularly. It’s easy to forget a long or complex passphrase if you only use it infrequently.

   **Tip:** We recommend reviewing your passphrase (e.g. by ensuring that you can log in to your SecureDrop account) on at least a monthly basis.

4. **Do not** use your passphrase anywhere else.

   If you use your SecureDrop passphrase on another system, a compromise of that system could theoretically be used to compromise SecureDrop. You should avoid reusing passphrases in general, but it is especially important to avoid doing so in the context of SecureDrop.
4.2 For Sources

Your passphrase is associated with your pseudonymous account and all of your activity on the SecureDrop server. In order to preserve your anonymity, you should avoid creating physical or digital associations between yourself and your passphrase as much as possible.

4.3 For Journalists/Admins

While Sources only have one passphrase that they are required to manage, Journalists and Admins unfortunately have to manage a veritable menagerie of credentials.

We have tried to minimize the number of credentials that Journalists and admins actually have to remember by automating the storage and entry of credentials on the Tails workstations wherever possible. For example, shortcut icons are created on the Desktop of each Tails workstation to make it easy to access the Tor Onion Services without having to look up their .onion addresses every time.

Ideally, each admin would only have to:
1. Keep track of their Admin Workstation Tails USB.
2. Remember the passphrase to unlock the persistent storage on that Tails USB.

And each Journalist would only have to:
1. Keep track of their Journalist Workstation Tails USB.
2. Keep track of their Secure Viewing Station Tails USB (and the associated Secure Viewing Station computer).
3. Remember the passphrases to unlock the persistent storage on both of these Tails USBs.

Memorizing further passphrases beyond the ones listed above is counterproductive: an attacker with access to any of those environments would be able to pivot to anything they wish to access, and increasing the burden of keeping track of additional credentials is unpleasant for journalists and admins and increases the risk that they will either forget their credentials, compromising the availability of the system, or compensate for the difficulty by using weak or reused credentials, potentially compromising the security of the system.

There is a detailed list of the credentials that must be managed by each end user role in Passphrases. We recommended using the KeePassX password manager included in Tails to store your credentials and minimize the passphrases that you need to memorize to just the passphrases for the persistent storage on your Tails USBs.

For the Transfer Device and the Export Device, which are used to copy files to and from the air-gapped Secure Viewing Station, we recommend using encrypted USB drives with passphrases stored in the journalist’s own password manager (preferably one which is accessible on their smartphone). This ensures that the journalist will have quick access to these passphrases when they need them.

If your organization is not using a password manager already, please see the Freedom of the Press Foundation guide to choosing one.
SecureDrop is an open-source whistleblower submission system that media organizations can use to securely accept
documents from and communicate with anonymous sources. It was originally created by the late Aaron Swartz and is
currently managed by Freedom of the Press Foundation.

Tip: Check out What makes SecureDrop Unique to read more about SecureDrop’s approach to keeping sources safe.

5.1 Technical Summary

SecureDrop is a tool for sources to communicate securely with journalists. The SecureDrop application environment
consists of three dedicated computers:

- **Secure Viewing Station:** A physically-secured and air-gapped laptop running the Tails operating system from
  a USB stick, that journalists use to decrypt and view submitted documents.

- **Application Server:** An Ubuntu server running two segmented Tor hidden services. The source connects to the
  Source Interface, a public-facing Tor Onion Service, to send messages and documents to the journalist. The
  journalist connects to the Journalist Interface, an authenticated Tor Onion Service, to download encrypted
documents and respond to sources.

- **Monitor Server:** An Ubuntu server that monitors the Application Server with OSSEC and sends email alerts.

These computers should all physically be in your organization’s office.

In addition to these dedicated computers, each journalist will also need a computer to connect to SecureDrop:

- **Journalist Workstation:** The computer used by the journalist to connect to the Journalist Interface to download
  encrypted documents that they will transfer to the Secure Viewing Station. The Journalist Workstation is
  also used to respond to sources via the Journalist Interface.

Depending on the news organization’s threat model, the Journalist Workstation can either be the journalist’s every-day
laptop or a dedicated computer. In either case, it is recommended that journalists always use the Tails operating system
on their Journalist Workstation when connecting to the Journalist Interface.
SecureDrop administrators will also require a computer to connect to SecureDrop and perform administrative tasks via SSH or the *Journalist Interface*. This computer is referred to as the *Admin Workstation*, and must be capable of running the *Tails* operating system. The *Admin Workstation* may also be used as a *Journalist Workstation* if necessary.

**Note:** The SecureDrop installation guide includes documentation on setting up Tails-based *Admin Workstation* and *Journalist Workstation* USB sticks. It is strongly recommended that these be used in preference to other undocumented solutions.

### 5.2 Infrastructure

There are four main components of SecureDrop: the servers, the admins, the sources, and the journalists.

#### 5.2.1 Servers

At SecureDrop’s heart is a pair of servers: the *Application ("App") Server*, which runs the core SecureDrop software, and the *Monitor ("Mon") Server*, which keeps track of the *Application Server* and sends out alerts if there’s a problem. These two servers run on dedicated hardware connected to a dedicated firewall appliance. They are typically located physically inside the newsroom.

#### 5.2.2 Admins

The SecureDrop servers are managed by a systems admin; for larger newsrooms, there may be a team of systems admins. The admin uses a dedicated *Admin Workstation* running *Tails*, connects to the *Application* and *Monitor Servers* over authenticated *Tor Onion Services*, and manages them using *Ansible*. 
5.2.3 Sources

A source submits documents and messages by using Tor Browser (or Tails) to access the Source Interface: a public Tor Hidden Service. Submissions are encrypted in place on the Application Server as they are uploaded.

5.2.4 Journalists

Journalists working in the newsroom use two machines to interact with SecureDrop. First, they use a Journalist Workstation running Tails to connect to the Journalist Interface, an authenticated Tor Hidden Service. Journalists download GPG-encrypted submissions and copy them to a Transfer Device (a thumb drive or DVD). Those submissions are then connected to the airgapped Secure Viewing Station (SVS) which holds the key to decrypt them. Journalists can then use the SVS to read, print, and otherwise prepare documents for publication. Apart from those deliberately published, decrypted documents are never accessed on an Internet-connected computer.

Note: The terms in italics are terms of art specific to SecureDrop. The Glossary provides more-precise definitions of these and other terms. SecureDrop is designed against a comprehensive Threat Model, and has a specific notion of the roles that are involved in its operation.

5.3 Operation

5.3.1 Planning & Preparation

Setting up SecureDrop is a multi-step process. Before getting started, you should make sure that you’re prepared to operate and maintain it. You’ll need a systems admin who’s familiar with Linux, the GNU utilities, and the Bash shell. You’ll need the hardware on which SecureDrop runs — this will normally cost $2000-$3000. The journalists in your organization will need to be trained in the operation of SecureDrop, and you’ll need to publish and promote your new SecureDrop instance afterwards — using your existing websites, mailing lists, and social media.

It is recommended that you have all of this planned out before you get started. If you need help, contact the Freedom of the Press Foundation who will be glad to help walk you through the process and make sure that you’re ready to proceed.

5.3.2 Technical Setup

Once you are familiar with the architecture and have all the hardware, setting up SecureDrop will take at least a day’s work for your admin. We recommend that you set aside at least a week to complete and test your setup.

5.3.3 Provisioning & Training

Once SecureDrop is installed, journalists will need to be provided with accounts, two-factor tokens, workstations, and so on — and then trained to use these tools safely and reliably. You will probably also need to train additional backup admins so that you can be sure that your SecureDrop setup keeps running even when your main admin is on holiday.

Introducing staff to SecureDrop takes half a day. Training a group to use SecureDrop proficiently takes at least a day — and a single trainer can only work with so many people at once. You will probably need to run several training sessions to instruct an entire newsroom. Depending on staff availability, training and provisioning may take a week or more. If you have multiple offices, training will need to happen at each location. Again, the Freedom of the Press Foundation are happy to help you plan and train your team.
5.3.4 Going Public

Once you have a SecureDrop instance and your team knows how to use it, you should test it thoroughly and then tell the world. The Freedom of the Press Foundation are happy to help you check that your SecureDrop setup is up-to-code and properly grounded. After that you’ll want to check out the best practices for your SecureDrop Landing Page and our guide to promoting your SecureDrop instance.
A number of terms used in this guide, and in the SecureDrop workflow diagram, are specific to SecureDrop. The list below attempts to enumerate and define these terms.

### 6.1 Source

The *Source* is the person who submits documents to SecureDrop and may use SecureDrop to communicate with a *Journalist*. A *Source* will always access SecureDrop through the *Source Interface* and must do so using Tor.

Instructions for using SecureDrop as a *Source* are available in our Source Guide.

### 6.2 Journalist

The *Journalist* uses SecureDrop to communicate with and download documents submitted by the *Source*. Journalists do this by using the *Journalist Workstation* to connect to the *Journalist Interface* through Tor.

The *Journalist* also uses a *Transfer Device* to move documents to the *Secure Viewing Station*. If a *Journalist* chooses to release any of these documents, they can be prepared for publication on the *Secure Viewing Station* before being transferred to an Internet-connected computer.

Instructions for using SecureDrop as a *Journalist* are available in our Journalist Guide.

### 6.3 Application Server

The *Application Server* runs the SecureDrop application. This server hosts both the website that sources access (the *Source Interface*) and the website that journalists access (the *Journalist Interface*). Sources, journalists, and admins may only connect to this server using Tor.
6.4 Monitor Server

The Monitor Server keeps track of the Application Server and sends out an email alert if something seems wrong. Only system admins connect to this server, and they may only do so using Tor.

6.5 Landing Page

The Landing Page is the public-facing webpage for a SecureDrop instance. This page is hosted as a standard (i.e. non-Tor) webpage on the news organization’s site. It provides first instructions for potential sources.

6.6 Source Interface

The Source Interface is the website that sources will access to submit documents and communicate with journalists. This site is hosted on the Application Server and can only be accessed through Tor.

Instructions for using the Source Interface are available in our Source Guide.

6.7 Journalist Interface

The Journalist Interface is the website that journalists access to download new documents and communicate with sources. This site is hosted on the Application Server and can only be accessed over Tor. In previous releases, this was called the Document Interface, but we have renamed it to avoid ambiguity.

Instructions for using the Journalist Interface are available in our Journalist Guide.

6.8 Journalist Workstation

The Journalist Workstation is a machine that is online and used together with the Tails operating system on the online USB stick. This machine will be used to connect to the Journalist Interface, download documents, and move them to the Secure Viewing Station using the Transfer Device.

Instructions for using the Journalist Workstation are available in our Journalist Guide.

6.9 Admin Workstation

The Admin Workstation is a machine that the system admin can use to connect to the Application Server and the Monitor Server using Tor and SSH. The admin will also need to have an Android or iOS device with the FreeOTP app installed.

6.10 Secure Viewing Station

The Secure Viewing Station (or SVS for short) is the computer you use to decrypt and view documents and messages submitted to your SecureDrop. This computer is permanently kept offline. It is “air-gapped”, meaning that there is a gap between it and any computer connected to the Internet.
You will boot the SVS from a designated USB stick running the Tails operating system. Once you have created it, you should never attach this USB stick to any Internet-connected device.

During the installation, the SVS is used to generate the Submission Key for encrypting and decrypting documents and messages submitted to SecureDrop. In addition, we recommend importing the public keys of individual journalists to the SVS, so you can securely encrypt files to their keys before exporting them.

Since this machine will never touch the Internet or run an operating system other than Tails on a USB, it does not need a hard drive or network device. We recommend physically removing the drive and any networking cards (wireless, Bluetooth, etc.) from this machine.

### 6.11 Submission Key

The Submission Key is the GPG keypair used to encrypt and decrypt documents and messages sent to your SecureDrop. Because the public key and private key must be treated very differently, we sometimes refer to them explicitly as the Submission Public Key and the Submission Private Key.

The Submission Public Key is uploaded to your SecureDrop servers as part of the installation process. Once your SecureDrop is online, anyone will be able to download it.

The Submission Private Key should never be accessible to a computer with Internet connectivity. Instead, it should remain on the Secure Viewing Station and on offline backup storage.

### 6.12 Two-Factor Authenticator

There are several places in the SecureDrop architecture where two-factor authentication is used to protect access to sensitive information or systems. These instances use the standard TOTP and/or HOTP algorithms, and so a variety of devices can be used to provide two-factor authentication for devices. We recommend using one of:

- FreeOTP for Android or for iOS installed
- A YubiKey

**Tip:** We recommend using FreeOTP (available for Android and for iOS) to generate two-factor authentication tokens because it is Free Software. However, if it does not work for you for any reason, alternatives exist:

- Google Authenticator for Android and iOS (proprietary)
- authenticator for the desktop (Free Software)

### 6.13 Transfer Device

The Transfer Device is the physical media (e.g., designated USB drive) used to transfer encrypted documents from the Journalist Workstation to the Secure Viewing Station, where they can be decrypted.

Please see the detailed security recommendations for the choice, configuration and use of your Transfer Device in the journalist guide and in the setup guide.
6.14 Export Device

The *Export Device* is the physical media (e.g., designated USB drive) used to transfer decrypted documents from the *Secure Viewing Station* to a journalist’s everyday workstation, or to another computer for additional processing.

Please see the detailed security recommendations for the choice, configuration and use of your *Export Device* in the *journalist guide* and in the *setup guide*.
CHAPTER 7

Passphrases

Each individual with a role (admin or journalist) at a given SecureDrop instance must generate and retain a number of strong, unique passphrases. The document is an overview of the passphrases, keys, two-factor secrets, and other credentials that are required for each role in a SecureDrop installation.

Note: We encourage each end user to use KeePassX, an easy-to-use password manager included in Tails, to generate and retain strong and unique passphrases. The SecureDrop code repository includes a template that you can use to initialize this database for an Admin Workstation or a Journalist Workstation. For more information, see the KeePassX setup instructions.

Tip: For best practices on managing passphrases, see Passphrase Best Practices.

7.1 Admin

The admin will be using the Admin Workstation with Tails to connect to the Application Server and the Monitor Server using Tor and SSH. The tasks performed by the admin will require the following set of credentials and passphrases:

- A passphrase for the persistent volume on the Admin Live USB.
- Additional credentials, which we recommend adding to Tails’ KeePassX password manager during the installation:
  - The Application Server and Monitor Server admin username and password (required to be the same for both servers).
  - The network firewall username and password.
  - The SSH private key and, if set, the key’s passphrase.
  - The GPG key that OSSEC will encrypt alerts to.
The admin’s personal GPG public key, if you want to potentially encrypt sensitive files to it for further analysis.

– The account details for the destination email address for OSSEC alerts.

– The Onion Services values required to connect to the Application and Monitor Servers.

The admin will also need to have a way to generate two-factor authentication tokens.

**Tip:** We recommend using FreeOTP (available for Android and for iOS) to generate two-factor authentication tokens because it is Free Software. However, if it does not work for you for any reason, alternatives exist:

- Google Authenticator for Android and iOS (proprietary)
- authenticator for the desktop (Free Software)

And the admin will also have the following two credentials:

- The secret code for the Application Server’s two-factor authentication.
- The secret code for the Monitor Server’s two-factor authentication.

### 7.2 Journalist

The journalist will be using the Journalist Workstation with Tails to connect to the Journalist Interface. The tasks performed by the journalist will require the following set of passphrases:

- A master passphrase for the persistent volume on the Tails device.
- A master passphrase for the KeePassX password manager, which unlocks passphrases to:
  - The Hidden Service value required to connect to the Journalist Interface.
  - The Journalist Interface.

The journalist will also need to have a two-factor authenticator, such as an Android or iOS device with FreeOTP installed, or a YubiKey. This means the journalist will also have the following credential:

- The secret code for the Journalist Interface’s two-factor authentication.

#### 7.2.1 Secure Viewing Station

The journalist will be using the Secure Viewing Station with Tails to decrypt and view submitted documents. The tasks performed by the journalist will require the following passphrases:

- A master passphrase for the persistent volume on the Tails device.

The backup that is created during the installation of SecureDrop is also encrypted with the application’s GPG key. The backup is stored on the persistent volume of the Admin Live USB.

#### 7.2.2 Transfer Device and Export Device

As noted in the setup guide, we recommend using encrypted USB drives for transferring files to the Secure Viewing Station, and for exporting them from the SVS in situations where using a secure printer or a similar analog conversion is not an option.

For every copy operation, the user will need to enter the USB drive’s encryption passphrase at least twice (on the computer they’re copying from, and on the computer they’re copying to). To make it easy for them to find the
passphrase, we recommend storing it in the journalist’s own existing password manager, which should be accessible using their smartphone.
This document outlines the required hardware components necessary to successfully install and operate a SecureDrop instance, and recommends some specific components that we have found to work well. If you have any questions, please email securedrop@freedom.press.

8.1 Hardware Overview

For an installation of SecureDrop, you must acquire:

- 2 computers with memory and hard drives to use as the SecureDrop servers.
- Mouse, keyboard, monitor (and necessary dongle or adapter) for installing the servers.
- At least 2 dedicated physical computers that can boot to Tails: one computer for the Secure Viewing Station, and one or more computers for the Admin Workstation(s)/Journalist Workstation(s).
- Dedicated airgapped hardware for the mouse, keyboard, and monitor (only if you are using a desktop for the Secure Viewing Station).
- Network firewall.
- At least 3 ethernet cables.
- Plenty of USB sticks: 1 drive for the master Tails stick, 1 drive for each Secure Viewing Station, 1 drive for each Transfer Device, 1 drive for each Export Device, and 1 drive for each admin and journalist.

Additionally, you may want to consider the following purchases:

- a printer without wireless network support, to use in combination with the Secure Viewing Station.
- an external hard drive to expand the storage capacity of the Secure Viewing Station.
- an external hard drive for server backups.
- a USB drive to store backups of your Tails workstation drives.
- a network switch, if you use a firewall with fewer than four ports.
• a hardware token for HOTP authentication, such as a YubiKey, if you want to use hardware-based two-factor authentication instead of a mobile app.

• a USB drive with a physical write protection switch, or a USB write blocker, if you want to mitigate the risk of introducing malware from your network to your Secure Viewing Station during repeated use of an Export Device.

• CD-R/DVD-R writers, if you want to use CD-Rs/DVD-Rs as transfer or export media, and a CD shredder that can destroy media consistent with your threat model.

In the sections that follow, we provide additional details on most of these items.

Tip: While a printer is not required, we highly recommend it. Printing documents is generally far safer than copying them in digital form. See our guide to working with documents for more information.

8.2 Advice for users on a tight budget

If you cannot afford to purchase new hardware for your SecureDrop instance, we encourage you to consider repurposing existing hardware to use with SecureDrop. If you are comfortable working with hardware, this is a great way to set up a SecureDrop instance for cheap.

Since SecureDrop’s throughput is significantly limited by the use of Tor for all connections, there is no need to use top of the line hardware for any of the servers or the firewall. In our experience, relatively recent recycled Dell desktops or servers are adequate for the SecureDrop servers, and recycled ThinkPad laptops work well for the Admin Workstation/Journalist Workstation.

Please note that very old laptops or desktops may not work for the workstations. Since the release of Tails 3.0, 32-bit computers are no longer supported.

If you choose to use recycled hardware, you should of course consider whether or not it is trustworthy; making that determination is outside the scope of this document.

8.3 Required Hardware

8.3.1 Servers

These are the core components of a SecureDrop instance.

• Application Server: 1 physical server to run the SecureDrop web services.

• Monitor Server: 1 physical server which monitors activity on the Application Server and sends email notifications to an admin.

• Network Firewall: 1 physical computer that is used as a dedicated firewall for the SecureDrop servers.

An acceptable alternative that requires more technical expertise is to configure an existing hardware firewall.

We are often asked if it is acceptable to run SecureDrop on cloud servers (e.g. Amazon EC2, DigitalOcean, etc.) or on dedicated servers in third-party datacenters instead of on dedicated hardware hosted in the organization. This request is generally motivated by a desire for cost savings and/or convenience. However: we consider it critical to have dedicated physical machines hosted within the organization for both technical and legal reasons:

• While the documents are stored encrypted at rest (via PGP) on the SecureDrop Application Server, the documents hit server memory unencrypted (unless the source used the GPG key provided to encrypt the documents first before submitting), and are then encrypted in server memory before being written to disk. If the machines
are compromised then the security of source material uploaded from that point on cannot be assured. The ma-
chines are hardened to prevent compromise for this reason. However, if an attacker has physical access to the
servers either because the dedicated servers are located in a datacenter or because the servers are not dedicated
and may have another virtual machine co-located on the same server, then the attacker may be able to com-
promise the machines. In addition, cloud servers are trivially accessible and manipulable by the provider that
operates them. In the context of SecureDrop, this means that the provider could access extremely sensitive in-
formation, such as the plaintext of submissions or the encryption keys used to identify and access the Tor Onion
Services.

- In addition, attackers with legal authority such as law enforcement agencies may (depending on the jurisdiction)
  be able to compel physical access, potentially with a gag order attached, meaning that the third party hosting
  your servers or VMs may be legally unable to tell you that law enforcement has been given access to your
  SecureDrop servers.

One of the core goals of SecureDrop is to avoid the potential compromise of sources through the compromise of
third-party communications providers. Therefore, we consider the use of virtualization for production instances of
SecureDrop to be an unacceptable compromise and do not support it. Instead, dedicated servers should be hosted in a
physically secure location in the organization itself. While it is technically possible to modify SecureDrop’s automated
installation process to work on virtualized servers (for example, we do so to support our CI pipeline), doing so in order
to run it on cloud servers is at your own risk and without our support or consent.

### 8.3.2 Workstations

These components are necessary to do the initial installation of SecureDrop and to process submissions using the
air-gapped workflow.

**Secure Viewing Station (SVS)**

1 physical computer used as an air-gap to decrypt and view submissions retrieved from the Application Server.

The chosen hardware should be solely used for this purpose and should have any wireless networking hardware
removed before use.

**Admin/Journalist Workstation(s)**

*At least 1* physical computer that is used as a workstation for SecureDrop admins and/or journalists.

Each Admin and Journalist will have their own bootable Tails USB with an encrypted persistent partition that they
will use to access SecureDrop. You will need at least one workstation to boot the Tails USBs, and may need more
depending on: the number of admins/journalists you wish to grant access to SecureDrop, whether they can share the
same workstation due to availability requirements, geographic distribution, etc.

### 8.3.3 USB Drive(s)

*At least 2* USB drives to use as a bootable Tails USB for the SVS and the Admin Workstation/Journalist Workstation.

If only one person is maintaining the system, you may use the same Tails instance as both the Admin Workstation and
the Journalist Workstation; otherwise, we recommend buying 1 drive for each admin and each journalist.

We also recommend buying an additional USB drive for making regular backups of your Tails workstations.

One thing to consider is that you are going to have a lot of USB drives to keep track of, so you should consider how
you will label or identify them and buy drives accordingly. Drives that are physically larger are often easier to label
(e.g. with tape, printed sticker or a label from a labelmaker).
8.3.4 Two-factor Authenticator

Two-factor authentication is used when connecting to different parts of the SecureDrop system. Each admin and each journalist needs a two-factor authenticator. We currently support two options for two-factor authentication:

- Your existing smartphone with an app that computes TOTP codes (e.g. FreeOTP for Android and for iOS).
- A dedicated hardware dongle that computes HOTP codes (e.g. a YubiKey).

**Tip:** We recommend using FreeOTP (available for Android and for iOS) to generate two-factor authentication tokens because it is Free Software. However, if it does not work for you for any reason, alternatives exist:

- Google Authenticator for Android and iOS (proprietary)
- authenticator for the desktop (Free Software)

8.3.5 Transfer Device(s) and Export Device(s)

Journalists need physical media (known as the Transfer Device) to transfer encrypted submissions from the Journalist Workstation to the Secure Viewing Station, to decrypt and view them there. If they deem a submission to be newsworthy, they may need physical media (known as the Export Device) to copy it to their everyday workstation.

Our standard recommendation is to use USB drives, in combination with volume-level encryption and careful data hygiene. Our documentation, including the journalist guide, is based on this approach. We also urge the use of a secure printer or similar analog conversions to export documents from the Secure Viewing Station, whenever possible.

You may want to consider enforcing write protection on USB drives when only read access is needed, or you may want to implement a workflow based on CD-Rs or DVD-Rs instead. We encourage you to evaluate these options in the context of your own threat model.

Please find some notes regarding each of these methods below, and see our recommendations in the setup guide for additional background.

**USB drives**

We recommend using one or multiple designated USB drives as the Transfer Device(s), and one or multiple designated USB drives as the Export Device(s). Whether one or multiple drives are appropriate depends on the number of journalists accessing the system, and on whether the team is distributed or not.

Our documentation explains how the Transfer Device can be encrypted using LUKS, and how the Export Device can be encrypted using VeraCrypt (which works across platforms). We have not evaluated hardware-based encryption options; if you do select a hardware solution, make sure that both devices work in Tails, and that the Export Device also works on the operating system(s) used by journalists accessing the Secure Viewing Station.

**USB drives with write protection (optional)**

When it is consistently applied and correctly implemented in hardware, write protection can prevent the spread of malware from the computers used to read files stored on a Transfer Device or an Export Device.

It is especially advisable to enable write protection before attaching an Export Device to an everyday workstation that lacks the security protections of the Tails operating system. For defense in depth, you may also want to enable write protection before attaching a Transfer Device to the Secure Viewing Station.

The two main options to achieve write protection of USB drives are:

- drives with a built-in physical write protection switch
• a separate USB write blocker device as used in forensic applications.

**DVD-Rs or CD-Rs**

Single-use, write-once media can be used to realize a transfer and export workflow that is always one-directional: files are transferred to the *Secure Viewing Station* and the media used to do so are destroyed; files are exported from the *Secure Viewing Station* and the media used to do so are destroyed.

If you want to realize such a workflow, we recommend purchasing separate drives for each computer that will write to or read from the media, to minimize the risks from malware compromising any one drive’s firmware.

You will also need a stack of blank DVD/CD-Rs, which you can buy anywhere, and a method to securely destroy media after use. Depending on your threat model, this can be very expensive; a cheap shredder can be purchased for less than $50, while shredders designed for use in Sensitive Compartmented Information Facilities (SCIFs) sell for as much as $3,000.

**8.3.6 Monitor, Keyboard, Mouse**

You will need these to do the initial installation of Ubuntu on the *Application* and *Monitor Servers*.

Depending on your setup, you may also need these to work on the *SVS*.

**8.4 Optional Hardware**

This hardware is not *required* to run a SecureDrop instance, but most of it is still recommended.

**8.4.1 Offline Printer**

We highly recommend purchasing a printer for your *Secure Viewing Station* and using it as the preferred method to make copies of documents received via SecureDrop.

By printing a submission, even a non-technical user can effectively mitigate many of the complex risks associated with malware or hidden metadata embedded in files received via SecureDrop. Your organization may also already have robust procedures in place for destroying sensitive printed documents.

**Important:** To maintain the integrity of the air-gap, this printer should be dedicated to use with the *Secure Viewing Station*, connected via a wired connection, and should not have any wireless communication capabilities.

While printing is notable for what it strips away, it is also important to remember what it preserves: QR codes or links that journalists may scan or type in; printer tracking information included in a scanned document; other visually encoded information. See the *Risks From Malware* section in the Journalist Guide for further guidance on this subject.

**8.4.2 Offline Storage**

The *SVS* is booted from a Tails USB drive, which has an encrypted persistent volume but typically has a fairly limited storage capacity since it’s just a USB drive. For installations that expect to receive a large volume of submissions, we recommend buying an external hard drive that can be used to store submissions that have been transferred from the *Application Server* to the *SVS*. 
Important: Like all storage media associated with SecureDrop, this drive should be encrypted and protected with a secure passphrase. We recommend using the tools built into Tails to encrypt the drive using LUKS.

If you are planning to use hardware RAID and/or hardware-based encryption, we recommend that you research Tails compatibility before a procurement decision.

8.4.3 Backup Storage

It’s useful to run periodic backups of the servers in case of failure. We recommend buying an external hard drive to store server backups.

Because this drive will be connected to the Admin Workstation to perform backups, it should not be the same drive used for Offline Storage.

Important: Like all storage media associated with SecureDrop, this drive should be encrypted and protected with a secure passphrase. We recommend using the tools built into Tails to encrypt the drive using LUKS.

If you are planning to use hardware RAID and/or hardware-based encryption, we recommend that you research Tails compatibility before a procurement decision.

8.4.4 Network Switch

If you follow our firewall recommendations, you do not need to purchase a switch.

If you use a firewall with fewer than four ports, you will need an additional Ethernet switch to perform installation and maintenance tasks with the Admin Workstation without disconnecting one of your servers.

8.4.5 Labeling Equipment

As you have probably noticed by now, a SecureDrop installation has a plethora of components. Some of these components can be hard to tell apart; for example, if you buy 3 of the same brand of USB sticks to use for the Admin Workstation, Journalist Workstation, and Secure Viewing Station, they will be indistinguishable from each other unless you label them. We recommend buying some labeling equipment up front so you can label each component as you provision it during the installation process.

There is a multitude of options for labeling equipment. We’ve had good results with small portable labelmakers, such as the Brother P-Touch PT-210 or the Epson LabelWorks LW-300. We like them because they produce crisp, easy-to-read labels, and it’s easy to customize the size of the label’s text, which is great for clearly labeling both large components (like computers) and small components (like USB sticks).

If you do not have a label maker available but have an inkjet printer available to you, it may also be possible to print and cut out labels using adhesive-backed paper and some scissors. These are some labels designed by our team which may be used for labeling:

- Admin Workstation Label
- Journalist Workstation Label
- Secure Viewing Station Label
- Firewall Label
- Application Server Label
8.5 Specific Hardware Recommendations

8.5.1 Application and Monitor Servers

We currently recommend the Intel NUC for SecureDrop servers.

**Note:** If using non-recommended hardware, ensure you remove as much extraneous hardware as physically possible from your servers. This could include: speakers, cameras, microphones, fingerprint readers, wireless, and Bluetooth cards.

**Intel 7th-gen NUC**

The Intel NUC (Next Unit of Computing) is an inexpensive, quiet, low-power device that can be used for the SecureDrop servers. There are a variety of models to choose from.

The NUCs typically come as kits, and some assembly is required. You will need to purchase the RAM and hard drive separately for each NUC and insert both into the NUC before it can be used. We recommend:

- 2x 240GB SSDs (2.5’’)
- 1x memory kit of compatible 2x4GB sticks - You can put one 4GB memory stick in each of the servers.

We have tested and can recommend the NUC7i5BNH - these tend to be readily available in retail stores.

The NUC7i5BNH has soldered-on wireless components, which cannot easily be removed. For security reasons, we recommend that you take the following steps to disable wireless functionality:

- before installation of the RAM and storage, disconnect the wireless antennae leads.
before the initial OS installation, boot into the BIOS by pressing F2 at startup, navigate to Advanced > Devices > Onboard Devices, and disable unwanted hardware - everything except LAN.
Other 7th-generation NUCs have also been reported to work, although we have not tested them. For example, the NUC7i5DNHE uses the same Ethernet chipset as the NUC7i5BNH, and also has a removable wireless card, simplifying the server setup process. However, it may be harder to find a retail source for this model.

8.5.2 Previous Server Recommendations
Intel 5th-gen NUC

We previously recommended the NUC5i5MYHE, however, it has now reached end-of-life. We will continue to support and test SecureDrop on this hardware, but if you are building a new SecureDrop instance we recommend using 7th-generation NUCs instead.

The NUC5i5MYHE supports wireless through optionally-purchased expansion cards. This means the wireless components aren’t soldered on which would make them nearly impossible to remove without inflicting damage to the NUC. This optional support is preferable, since you want neither WiFi nor Bluetooth.

**Note:** If you encounter issues booting Ubuntu on the NUCs, try updating the BIOS according to [these instructions](#).

**Caution:** Some older NUC BIOS versions will cause the server to brick itself if the device attempts to suspend. This has since been fixed in a BIOS update. See these release notes (PDF) for more details.

2014 Mac Minis

We previously recommended the 2014 Apple Mac Minis (part number MGEM2) for installing SecureDrop. These will soon be officially obsolete. Unfortunately the 2018 revision of the Mac Mini is not a viable candidate for use with SecureDrop, as security features of the device prevent Linux from being installed on its internal storage. We will continue to support existing instances using 2014 Mac Minis, but if you are building a new instance we recommend using the 7th-gen Intel NUCs.

2014 Mac Minis have removable wireless cards that you should remove. This requires a screwdriver for non-standard TR6 Torx security screws.

However, on the first install of Ubuntu Server the Mac Minis will not boot: this is a known and documented issue. The workaround requires a one-time modification after you install Ubuntu but before you move on to install SecureDrop. After Ubuntu is installed, for each Mac Mini you should:

1. Connect your Ubuntu installation media (USB drive or CD)
2. Boot your Mac Mini while holding down the **Option** key.
3. Select **EFI Boot** and select **Rescue a broken system** at the Ubuntu install screen.
4. Accept the default options for the install steps until you get to **Device to use as root file system**.
5. At the **Device to use as root file system** prompt, select `/dev/mon-vg/root` or `/dev/app-vg/root` for the monitor and application servers respectively.
6. Select to mount the separate `/boot` partition.
7. Select **Execute a shell in** `/dev/mon-vg/root` (or `/dev/app-vg/root`) and select **Continue**.
8. You should now be at a rescue Linux shell. Type `efibootmgr`, and you should see the following:

```
BootCurrent: 0000
Timeout: 5 seconds
BootOrder: 0080
Boot0000* ubuntu
Boot0080* Mac OS X
BootFFFF*
```

9. Type `efibootmgr -o 00`.
10. Again type `efibootmgr`. This time you should see the following:
11. Type `exit`.

12. Select **Reboot the system** and remove the installation media. Your server should now boot to Ubuntu by default.

### 8.5.3 Journalist Workstation and Admin Workstation

Both the *Journalist Workstation* and the *Admin Workstation* must be compatible with the Tails operating system. Compare any hardware you want to procure or allocate for this purpose against the list of known issues maintained by the Tails project, but please be advised that the list is far from exhaustive.

We advise against using Macs, as there are many Tails compatibility issues both with older and with newer models. Instead, we recommend the ThinkPad T series, and have had good experiences specifically with the T420 and T440. The **ThinkWiki** is an excellent, independently maintained resource for verifying general Linux compatibility of almost any ThinkPad model.

For any Tails workstation, we recommend at least 8GB of RAM.

### 8.5.4 Secure Viewing Station (SVS)

The *Secure Viewing Station* is a machine that is kept offline and only ever used together with the Tails operating system. This machine will be used to generate the GPG keys used by SecureDrop to encrypt submissions, as well as decrypt and view submissions. Since this machine will never touch the Internet or run an operating system other than Tails, it does not need a hard drive or network device; in fact, we recommend removing these components if they are already present.

As with the workstations, one good option is to buy a Linux-compatible laptop from the Lenovo ThinkPad T series. We have tested the T420 and successfully removed the wireless components with ease. It’s possible to re-purpose old laptops from other manufacturers, as long as the wireless components are removable.

Just as with the servers, you can also use an Intel NUC for the *SVS*. As noted before, NUCs do not ship with a hard drive, and can be configured without any wireless components, so you’ll save time by not having to remove these, since they won’t be present. However, NUCs do contain an IR receiver, which we recommend taping over with opaque masking tape.

If you choose to use an Intel NUC that differs from our recommended model, make sure you use one that offers wireless as an **option**. If the model is advertised as having “integrated wireless”, such as the **NUC5i5RYK**, this could mean it’s built into the motherboard, making it physically irremovable, and attempting to do so would risk damaging the unit; instead, look for attributes like **M.2 22x30 slot and wireless antenna pre-assembled (for wireless card support)**, as advertised by the **NUC5i5MYHE** that we recommend.

### 8.5.5 Tails USBs

We **strongly recommend** getting USB 3.0-compatible drives to run Tails from. The transfer speeds are significantly faster than USB 2.0, which means a live operating system booting from one will be much faster and more responsive.

You will need **at least** an 8GB drive to run Tails with an encrypted persistent partition. We recommend getting something in the 16-64GB range so you can handle large amounts of submissions without hassle. Anything more than that is probably overkill.
8.5.6 Transfer Device(s) and Export Device(s)

For USB drives with physical write protection, we have tested the Kanguru SS3 on Tails, and it works well with and without encryption.

If you want to use a setup based on CD-Rs or DVD-Rs, we’ve found the CDR/DVD writers from Samsung and LG to work reasonably well; you can find some examples here.

Please see our recommendations in the setup guide for additional background.

8.5.7 Network Firewall

We recommend the pfSense SG-3100. It has 3 NICs and an internal switch, increasing the number of available ports to 6.

8.5.8 Network Switch

This is optional, for people who are using a firewall with less than 4 ports. Any old switch with more than 3 ports will do, such as the 5-port Netgear ProSafe Ethernet Switch.

8.5.9 Printers

Careful consideration should be given to the printer used with the SVS. Most printers today have wireless functionality (WiFi or Bluetooth connectivity) which should be avoided because it could be used to compromise the air-gap.

Unfortunately, it is difficult to find printers that work with Tails, and it is increasingly difficult to find non-wireless printers at all. To assist you, we have compiled the following partial list of air-gap-safe printers that have been tested and are known to work with Tails:

<table>
<thead>
<tr>
<th>Printer Model</th>
<th>Testing Date</th>
<th>Tails Versions</th>
<th>Printer Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP DeskJet F4200</td>
<td>06/2017</td>
<td>3.0</td>
<td>Color Inkjet</td>
</tr>
<tr>
<td>HP DeskJet 1112</td>
<td>06/2017</td>
<td>3.0</td>
<td>Color Inkjet</td>
</tr>
<tr>
<td>HP DeskJet 1110</td>
<td>08/2017</td>
<td>3.1</td>
<td>Color Inkjet</td>
</tr>
<tr>
<td>HP LaserJet 400 M401n</td>
<td>06/2015</td>
<td>1.4</td>
<td>Monochrome Laser</td>
</tr>
<tr>
<td>HP DeskJet 6940</td>
<td>04/2015</td>
<td>1.3.2</td>
<td>Monochrome Inkjet</td>
</tr>
</tbody>
</table>

Note: We’ve documented both the HP DeskJet F4200 and HP LaserJet 400 M401n with screenshots of the installation process, in our section on Setting Up a Printer in Tails. While the F4200 installed automatically, the 400 M401n required that we set “Make and model” to “HP LaserJet 400 CUPS+Gutenprint v5.2.9” when manually configuring the drivers.

If you know of another model of printer that fits our requirements and works with Tails, please submit a pull request to add it to this list.

8.5.10 Monitor, Keyboard, Mouse

We don’t have anything specific to recommend when it comes to displays. You should make sure you know what monitor cable you need for the servers, since you will need to connect them to a monitor to do the initial Ubuntu installation.

You should use a wired (USB) keyboard and mouse, not wireless.
Before you get started, read the Overview and familiarize yourself with the glossary and the passphrases involved in SecureDrop’s operations. You may wish to leave these documents open in other tabs for reference as you work.

SecureDrop is a technical tool. It is designed to protect journalists and sources, but no tool can guarantee safety. This guide will instruct you in installing and configuring SecureDrop, but it does not explain how to use it safely and effectively. Put another way: at the end of this guide, you will have built a car; you will not know how to drive. The Deployment Guide contains best practices for working with SecureDrop. Make sure to read it after completing the installation.

Installing SecureDrop is an extended manual process which requires extensive preparation, equipment, and organizational support. You will likely need to set aside a day to complete the install process. Installing SecureDrop requires an admin with at least basic familiarity with Linux, the GNU core utilities, and Bash shell. If you are not proficient in these areas, it is strongly recommended that you contact the Freedom of the Press Foundation for installation assistance.

Before you begin, you will need to assemble the required hardware.

To assist in the installation process, we offer a SecureDrop Installation Worksheet. It is critical that you destroy this worksheet when your installation is complete and all of your passphrases have been safely stored in a password manager.

When running commands or editing configuration files that include filenames, version numbers, usernames, hostnames, or IP addresses, make sure to use the appropriate values for your instance.

Once you’re familiar with SecureDrop, have made your plan, ensured your organization is ready to follow through, and assembled the necessary hardware, you’re ready to begin.

**Warning:** Remember to destroy the SecureDrop Installation Worksheet after the installation is complete.
Create Tails USBs

Tails is a privacy-enhancing live operating system that runs on removable media, such as a DVD or a USB stick. It sends all your Internet traffic through Tor, does not touch your computer's hard drive, and securely wipes unsaved work on shutdown.

Most of the work of installing, administering, and using SecureDrop is done from computers using Tails, so the first thing you need to do is set up several USB drives with the Tails operating system. To get started, you’ll need two Tails drives: one for the Admin Workstation and one for the Secure Viewing Station. Later, you’ll set up a bunch more Tails drives for your journalists and backups, but for now you just need two.

As soon as you create a new Tails drive, label it immediately. USB drives all look alike and you’re going to be juggling a whole bunch of them throughout this installation. Label immediately. Always.

10.1 Install Tails

We recommend creating an initial Tails Live DVD or USB, and then using that to create additional Tails drives with the Tails Installer, a special program that is only available from inside Tails. All of your Tails drives will need persistence: a way of safely saving files and so on between reboots. It is only possible to set up persistence on USB drives which were created via the Tails Installer.

The Tails website has detailed and up-to-date instructions on how to download and verify Tails, and how to create a bootable Tails USB drive.

Important: Make sure you verify the Tails .iso using one of the methods described on the Tails website.

Follow the instructions at these links and then return to this page:

- Download and verify the Tails .iso
- Install onto a USB drive

You will need to create 3 Tails USBs to perform the SecureDrop installation:
1. A “master” Tails USB, which you will create by copying a Tails .iso onto a USB drive, using one of the techniques outlined in the Tails documentation. This Tails USB is only used for creating other Tails USBs with the Tails Installer.

2. The Secure Viewing Station Tails USB.

3. The Admin Workstation Tails USB.

**Tip:** This process will take some time, most of which will be spent waiting around. Once you have the “master” copy of Tails, you have to boot it, create another Tails drive with the Tails Installer, shut down, and boot into the new Tails USB to complete the next step of setting up the persistence - for each additional Tails USB.

**Note:** Tails doesn’t always completely shut down and reboot properly when you click “restart”, so if you notice a significant delay, you may have to manually power off and restart your computer for it to work properly.

## 10.2 Enable Persistent Storage

Creating an encrypted persistent volume will allow you to securely save information and settings in the free space that is left on your Tails drive. This information will remain available to you even if you reboot Tails. (Tails securely erases all other data on every shutdown.)

You will need to create a persistent storage on each Tails drive, with a unique passphrase for each.

Please use the instructions on the Tails website to make the persistent volume on each Tails drive you create. When creating the persistence volume, you will be asked to select from a list of features, such as ‘Personal Data’. You should enable all features by selecting each item in the list.

Some other things to keep in mind:

- Right now, you need to create a persistent volume on both the Admin Workstation Tails drive and the Secure Viewing Station Tails drive.

- Each Tails persistent volume should have an unique and complex passphrase that’s easy to write down or remember. We recommend using Diceware passphrases.

- Each journalist will need their own Tails drive with their own persistent volume secured with their own passphrase — but that comes later.

- Journalists and admins will eventually need to remember these passphrases. We recommend using spaced-repetition to memorize Diceware passphrases.

**Warning:** Make sure that you never use the Secure Viewing Station Tails drive on a computer connected to the Internet or a local network. This Tails drive will only be used on the air-gapped Secure Viewing Station.
Set Up the Secure Viewing Station

The Secure Viewing Station is the computer where journalists read and respond to SecureDrop submissions. Once submissions are encrypted on the Application Server, only the Secure Viewing Station has the key to decrypt them. The Secure Viewing Station is never connected to the internet or a local network, and only ever runs from a dedicated Tails drive. Journalists download encrypted submissions using their Journalist Workstation, copy them to a Transfer Device (a USB drive or a DVD) and physically transfer the Transfer Device to the Secure Viewing Station.

Since the Secure Viewing Station never uses a network connection or an internal hard drive, we recommend that you physically remove any internal storage devices or networking hardware such as wireless cards or Bluetooth adapters. If the machine has network ports you can’t physically remove, you should clearly cover these ports with labels noting not to use them. For an even safer approach, fill a port with epoxy to physically disable it. We also recommend you remove the speakers from the device (or just cut the audio cables if that’s easier). This is to prevent exfiltration of data from the airgap via ultrasonic audio, which cannot be heard by humans. If you have questions about repurposing hardware for the Secure Viewing Station, contact the Freedom of the Press Foundation.

You should have a Tails drive clearly labeled “SecureDrop Secure Viewing Station”. If it’s not labeled, label it right now, then boot it on the Secure Viewing Station. After it loads, you should see a “Welcome to Tails” screen, also known as the Tails Greeter.

Enter your passphrase to unlock the persistent storage, then press Unlock. Before starting Tails, set an administration password for use with this Tails session. To do so, click the + button under “Additional Settings”. Click Administration Password in the list of settings. Enter the password twice, click Add, then click Start Tails.

Note: The Tail administration password is a one-time password. It is reset every time you shut down Tails. Pick a password you will be able to remember for the length of this session.

We will now prepare the Secure Viewing Station.

11.1 Ensure Filenames are Preserved

In order to preserve filenames when you decrypt submissions, on each Secure Viewing Station, you should open a Terminal and type the following commands:
SecureDrop Documentation, Release 1.0.0

```
cd /live/persistence/TailsData_unlocked/dotfiles
echo "/usr/bin/dconf write /org/gnome/nautilus/preferences/automatic-decompression
˓false" > .xsessionrc
```

**Note:** This only needs to be done once on each Secure Viewing Station. After a reboot it will persist.

### 11.2 Correct the System Time

After booting up Tails on the Secure Viewing Station, you will need to manually set the system time before you create the SecureDrop Submission Key. This operation requires the Tails administration password to be set (see above).

To set the system time:

1. Click the upper right down arrow in the menu bar and select the wrench icon:
2. Then click **Date & Time**.

3. Click **Unlock**. Type in the admin password you set when you started up Tails.

4. Set the correct time, region and city.

5. Click **Lock**, exit Settings and wait for the system time to update in the top panel.
CHAPTER 12

Set Up the Transfer Device and the Export Device

Journalists copy submissions from their Journalist Workstation to the Secure Viewing Station using the Transfer Device.

For exporting submissions from the Secure Viewing Station, we recommend using a secure printer or a similar analog conversion process wherever possible. For cases where an electronic file transfer is necessary, we recommend setting up an Export Device, separate from the Transfer Device.

Important: Understand the security risks of working with files in digital form

After downloading a submission on the Journalist Workstation, a journalist will copy it to the Transfer Device and carry it to the air-gapped Secure Viewing Station to decrypt and review it. If the journalist then copies the decrypted file in its original form to an Internet-connected computer, they may expose themselves, their colleagues, or their sources to significant risks, e.g.:

- A submission may be infected with malware targeting your newsroom.
- If your Secure Viewing Station has not been updated in a while, it may have software vulnerabilities an attacker can exploit, e.g., to exfiltrate the Submission Private Key alongside a legitimate submission.
- The submission may contain metadata identifying the source which has not yet been cleaned up.

These risks are not specific to SecureDrop. They’re inherent to dealing with tips sent in digital form.

This is why we place the strongest emphasis on always picking the most secure available export method for a given submission. Printing documents or re-recording audio and video files can eliminate most categories of malware and metadata (QR code malware and tracking dots being the most notable exceptions).

If and when you do need to copy decrypted files in electronic form, the recommendations below are intended to establish a baseline of security. Please consider these recommendations in the context of your own threat model, and do not hesitate to contact us via securedrop@freedom.press (GPG encrypted) if we can help.
12.1 Choose media types and encryption

You will need to decide what storage media to use for the Transfer Device and the Export Device, and which encryption scheme to apply to each device. There are many options to consider: USB flash drives, write-once media like CD-Rs and DVD-Rs, external hard drives, and so on.

The following recommendation is intended to balance security, usability and cost considerations, and you may want to modify it based on your threat model:

- Use USB flash drives for both the Transfer Device and the Export Device.
- Encrypt the Transfer Device using LUKS, which works in the Tails environment and in other Linux environments.
- Encrypt the Export Device using VeraCrypt, which works across platforms.

If you follow this recommendation, it is important that the contents of the Transfer Device and the Export Device are always wiped after a copy operation is completed.

Note: You may want to purchase a USB device with a physical write protection switch for the Export Device, to enforce write protection whenever it is attached to an Internet-connected everyday workstation. This ensures that malware cannot spread from infected computers in your network to the Secure Viewing Station.

Another option is to purchase a hardware USB write blocker as used in forensics, and enforce its usage whenever the Export Device is attached to an Internet-connected workstation.

Write-once media like CDs and DVDs can be a reasonable alternative to this setup. If you implement a workflow based on CDs or DVDs, it is crucial that they are destroyed immediately after use. While you can find CD/DVD destroyers at a relatively low cost, those built to the highest standards of security sell in the $2,500 to $3,000 price range as of 2019.

12.2 Decide how to manage encryption passphrases

Because files are copied between multiple computers, KeePassX in Tails is not necessarily the most convenient option for managing the encryption passphrases for your Transfer Device or your Export Device. While Tails itself gives you the option to “remember” passphrases, this option does not work across reboots.

A simple alternative is to make sure that every journalist stores the Transfer Device and Export Device passphrases in their own password manager, which ideally will synchronize to their smartphone. See the Freedom of the Press Foundation guide for choosing a password manager if you are not currently using one.

Tip: The user will have to enter this passphrase repeatedly. For this reason, we recommend using diceware instead of random character sequences that are difficult to type. If your password manager does not support generating diceware passphrases, see the EFF guide for information on how to do it yourself.

12.3 Create USB Transfer Device

The easiest and recommended option for a Transfer Device is a USB drive. If you have a large team of journalists you may want to create several of these. Here we’ll just walk through making one Transfer Device.

1 Tails screenshots were taken on Tails 3.0.1. Please make an issue on GitHub if you are using the most recent version of Tails and the interface is different from what you see here.
Note: This process will destroy all data currently on the drive.

First, label your USB drive “SecureDrop Transfer Device”.

On the Secure Viewing Station, open the Applications menu in the top left corner and select Utilities then Disks:

Connect your Transfer Device then pick your device in the menu on the left. Since we’re going to destroy all the data on this drive, it’s important that you pick the right drive. It should be named something that sounds similar to the
manufacturer’s label on the outside of the drive, and it will only appear after you plug it in. Double check that you have clicked on the correct drive:

<table>
<thead>
<tr>
<th>Drive Type</th>
<th>Model</th>
<th>Size</th>
<th>Partitioning</th>
<th>Serial Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>320 GB Hard Disk</td>
<td>General USB Flash Disk (1.00)</td>
<td>16 GB (15,640,600,576 bytes)</td>
<td>Master Boot Record</td>
<td>04744D000009061</td>
</tr>
<tr>
<td>CD/DVD Drive</td>
<td>v1.00</td>
<td>16 GB (15,640,600,576 bytes)</td>
<td>Master Boot Record</td>
<td>04744D000009061</td>
</tr>
<tr>
<td>16 GB Drive</td>
<td>General USB Flash Disk</td>
<td>16 GB (15,640,600,576 bytes)</td>
<td>Master Boot Record</td>
<td>04744D000009061</td>
</tr>
<tr>
<td>11 GB Loop Device</td>
<td>General USB Flash Disk</td>
<td>16 GB (15,640,600,576 bytes)</td>
<td>Master Boot Record</td>
<td>04744D000009061</td>
</tr>
</tbody>
</table>

Once you’re sure you have the right drive, click the interlocking gears, then **Format Partition**…. 

**Note:** If there are multiple existing partitions on the drive, you should first click the “-” icon on the left of the interlocking gears icon to delete each partition, and then create another partition that fills all free space with the options as shown below.
Give the partition on your *Transfer Device* a descriptive name like “Transfer Device” and select the options as in the following screenshot:
As noted earlier, we recommend storing this passphrase in the password manager for every user who will copy files using the Transfer Device, not in KeePassX. Because users will have to type in this passphrase every time they mount the Transfer Device in Tails, we recommend using a diceware passphrase.

After typing in the passphrase, click Format to continue. The Disks utility will ask you if you are sure: click Format to continue. After a few seconds, your new Transfer Device should be ready for use. If you haven’t already, make sure to label it.

12.4 Create a USB Export Device

We recommend using a fully encrypted USB drive for copying files off the Secure Viewing Station. This is even more important than for the Transfer Device, as the risk of accidentally leaving decrypted files on the Export Device is significant.

Because the Export Device will need to be mounted on both Tails and the journalist’s everyday workstation, you will need to use an encryption scheme that works on both operating systems.

We recommend the use of VeraCrypt. It is actively maintained cross-platform software that has been independently audited and is free to use.

VeraCrypt-encrypted media can be opened in the Tails operating system and on common Linux distributions without installing additional software. To open VeraCrypt media on Windows or Mac workstations, or to create VeraCrypt drives, you need to install the VeraCrypt software. The guide by Freedom of the Press Foundation provides instructions for encrypting storage media using VeraCrypt.
Keep in mind that each journalist using a Windows or Mac workstation will need to have the VeraCrypt software installed on their computer to access the encrypted Export Device.

**Note:** We recommend against installing the VeraCrypt software on the Journalist Workstation, the Admin Workstation or the Secure Viewing Station. The software installed in the persistent volume of these Tails drives should be kept to a minimum. You do not need to install the software to decrypt VeraCrypt drives on these workstations, and you can create them from another computer.

Larger organizations may want to consider setting up a controlled environment for creating VeraCrypt-encrypted Export Devices and providing them to journalists, to ensure that each drive is provisioned in a secure manner.

As with the Transfer Device, we recommend storing the passphrase in the password manager of each user who will use a given Export Device.

Hardware-encrypted USB drives can be a reasonable alternative to VeraCrypt. We cannot currently offer a specific recommendation, but please bear in mind that the drive must work across platforms (including Tails). We recommend selecting a vendor that has fully opened the source code and specifications of their devices and encouraged third party audits.

### 12.4.1 Limiting write access

If you re-use the same Export Device for multiple copy operations, there is the risk of introducing malware to the Secure Viewing Station from your network. Depending on your threat model, there are steps you may want to take to mitigate that risk.

One option is to restrict write access to the Export Device before it is plugged into a device other than the Secure Viewing Station. Some USB flash drives come with a physical write protection switch, and write blockers are used in forensics to ensure storage media are not modified during examination.

Full-size SD cards also come with physical write protection switches. However, this write protection is fully host-based (the host operating system can choose to ignore it), and should therefore be considered less secure against sophisticated malware.

**Tip:** For defense in depth, consider implementing a similar write protection strategy for the Transfer Device (enabling write protection before attaching the Transfer Device to the Secure Viewing Station).

Please see our [guide to working with documents](#) for additional recommendations regarding malware mitigation.

---

**12.4. Create a USB Export Device**
Generate the Submission Key

When a document or message is submitted to SecureDrop by a source, it is automatically encrypted with the Submission Key. The private part of this key is only stored on the Secure Viewing Station which is never connected to the Internet. SecureDrop submissions can only be decrypted and read on the Secure Viewing Station.

We will now generate the Submission Key. If you aren’t still logged into your Secure Viewing Station from the previous step, boot it using its Tails USB stick, with persistence enabled.

**Important:** Do not follow these steps before you have fully configured the Secure Viewing Station according to the instructions. The private key you will generate in the following steps is one of the most important secrets associated with your SecureDrop installation. This procedure is intended to ensure that the private key is protected by the air-gap throughout its lifetime.

### 13.1 Create the Key

1. Navigate to **Applications Terminal** to open a terminal.
2. In the terminal, run `gpg --full-generate-key:`
3. When it says **Please select what kind of key you want**, choose “(1) RSA and RSA (default)”.

4. When it asks **What keysize do you want?**, type **4096**.

5. When it asks **Key is valid for?**, press Enter. This means your key does not expire.

6. It will let you know that this means the key does not expire at all and ask for confirmation. Type **y** and hit Enter to confirm.
SecureDrop Documentation, Release 1.0.0

7. **Next it will prompt you for user ID setup. Use the following options:**

   - **Real name:** “SecureDrop”
   - **Email address:** leave this field blank
   - **Comment:** [Your Organization’s Name] SecureDrop Submission Key

8. GPG will confirm these options. Verify that everything is written correctly. Then type O for (O)kay and hit enter to continue:

```
amnesia@amnesia:~ $ gpg --full-generate-key
```

```
gpg (GnuPG) 2.1.18; Copyright (C) 2017 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) RSA (sign only)
   (4) RSA (sign only)
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
   0 = key does not expire
   <n> = key expires in n days
   <n>w = key expires in n weeks
   <n>m = key expires in n months
   <n>y = key expires in n years
Key is valid for? (0)
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: [Your Organization’s Name] SecureDrop Submission Key

```

13.1. Create the Key
9. A box will pop up (twice) asking you to type a passphrase. Since the key is protected by the encryption on the Tails persistent volume, it is safe to simply click **OK** without entering a passphrase.

10. The software will ask you if you are sure. Click **Yes, protection is not needed**.

11. Wait for the key to finish generating.

### 13.2 Export the Submission Public Key

To manage GPG keys using the graphical interface (a program called Seahorse), click the clipboard icon in the top right corner and select “Manage Keys”. Click “GnuPG keys” and you should see the key that you just generated.
1. Select the key you just generated and click “File” then “Export”.

2. Save the key to the Transfer Device as SecureDrop.asc, and make sure you change the file type from “PGP keys” to “Armored PGP keys” which can be switched at the bottom of the Save window. Click the ‘Export’ button after switching to armored keys.

**Note:** This is the public key only.
You’ll need to provide the fingerprint of this new key during the installation. Double-click on the newly generated key.
and change to the Details tab. Write down the 40 hexadecimal digits under Fingerprint.

Note: Your fingerprint will be different from the one in the example screenshot.

At this point, you are done with the Secure Viewing Station for now. You can shut down Tails, grab the Admin Workstation Tails USB and move over to your regular workstation.

13.2. Export the Submission Public Key
CHAPTER 14

Set up the Admin Workstation

Earlier, you should have created the Admin Workstation Tails USB along with a persistence volume for it. Now, we are going to add a couple more features to the Admin Workstation to facilitate SecureDrop’s setup.

If you have not switched to and booted the Admin Workstation Tails USB on your regular workstation, do so now.

14.1 Start Tails with Persistence Enabled

After you boot the Admin Workstation Tails USB on your normal workstation, you should see a Welcome to Tails screen with Encrypted Persistent Storage. Enter your password and click Unlock. Do NOT click Start Tails yet. Under Additional Settings click the plus sign.

Click Administration password, enter a password for use with this specific Tails session and click Add. And finally click Start Tails.

Note: The Administration password is a one-time password. It will reset every time you shut down Tails.

After Tails finishes booting, make sure you’re connected to the Internet and that the Tor status onion icon is not crossed out, consulting the icons in the upper right corner of the screen.

14.2 Download the SecureDrop repository

The rest of the SecureDrop-specific configuration is assisted by files stored in the SecureDrop Git repository. We’re going to be using this again once SecureDrop is installed, but you should download it now. To get started, open a terminal. You will use this Terminal throughout the rest of the install process.

Start by running the following commands to download the git repository.
cd ~/Persistent
git clone https://github.com/freedomofpress/securedrop.git

Note: Since the repository is fairly large and Tor can be slow, this may take a few minutes.

Caution: Do not download SecureDrop Git repository as a Zip file, or any other means. Only download by using the given git command.

14.2.1 Verify the Release Tag

Important: It is crucial for the integrity of your installation that you carefully follow the instructions below. By following these steps, you will verify if your copy of the codebase has been approved by the SecureDrop development team.

Download and verify the SecureDrop Release Signing Key using the following command:

```bash
gpg --keyserver hkps://keys.openpgp.org --recv-key "2224 5C81 E3BA EB41 38B3 6061 310F 5612 00F4 AD77"
```

If you are not copy-pasting this command, we recommend you double-check you have entered it correctly before pressing enter. GPG will implicitly verify that the fingerprint of the key received matches the argument passed.

If GPG warns you that the fingerprint of the key received does not match the one requested, do not proceed with the installation. If this happens, please contact us at securedrop@freedom.press.

Note: If the `--recv-key` command fails, first double-check that Tails is connected to Tor. Once you’ve confirmed that you’re successfully connected to Tor, try re-running the `--recv-key` command a few times.

If the command still fails, the keys.openpgp.org keyserver may be down. In that case, we recommend downloading the key from the SecureDrop website:

```bash
cd ~/Persistent
torify curl -LO https://securedrop.org/securedrop-release-key.asc
```

Before importing it, inspect the key’s fingerprint using the following command. The `--dry-run` option ensures that the key is not imported just yet:

```bash
gpg --with-fingerprint --import-options import-show --dry-run --import securedrop-release-key.asc
```

Compare the fingerprint in the output with the fingerprint at the beginning of this section. If the fingerprints match, you can safely import the key, using the following command:

```bash
gpg --import securedrop-release-key.asc
```

If you encounter any difficulties verifying the integrity of the release key, do not proceed with the installation. Instead, please contact us at securedrop@freedom.press.

Once you have imported the release key, verify that the current release tag was signed with the release signing key:
cd ~/Persistent/securedrop/
git checkout 1.0.0

git tag -v 1.0.0

You should see Good signature from "SecureDrop Release Signing Key" in the output of that last command along with the fingerprint above.

**Important:** If you do not see the message above, signature verification has failed and you should **not** proceed with the installation. If this happens, please contact us at securedrop@freedom.press.

### 14.3 Create the Admin Passphrase Database

We provide a KeePassX password database template to make it easier for admins and journalists to generate strong, unique passphrases and store them securely. Once you have set up Tails with persistence and have cloned the repo, you can set up your personal password database using this template.

You can find the template in `tails_files/securedrop-keepassx.kdbx` in the SecureDrop repository that you just cloned.

To use the template:

- Open the KeePassX program which is already installed on Tails
- Select **Database**, **Open database**, and navigate to the location of `securedrop-keepassx.kdbx`, select it, and click **Open**
- Check the **password** box and hit **OK**
- Click **Database** and **Save Database As**
- Save the database in the Persistent folder

**Tip:** If you would like to add a master password, navigate to **Database** and **Change master key**. Note that since each KeePassX database is stored on the encrypted persistent volume, this additional passphrase is not necessary.

**Warning:** You will not be able to access your passwords if you forget the master password or the location of the key file used to protect the database.

In case you wish to manually create a database, the suggested password fields in the template are:

**Admin:**

- Admin account username
- App Server SSH Onion URL
- Email account for sending OSSEC alerts
- Monitor Server SSH Onion URL
- Network Firewall Admin Credentials
- OSSEC GPG Key
SecureDrop Login Credentials

**Journalist:**
- Auth Value: Journalist Interface
- Onion URL: Journalist Interface
- Personal GPG Key
- SecureDrop Login Credentials

**Secure Viewing Station:**
- SecureDrop GPG Key

**Backup:**
- This section contains clones of the above entries in case a user accidentally overwrites an entry.
Now that you’ve set up your password manager, you can move on to setting up the Network Firewall. You should stay logged in to the Admin Workstation to access the Network Firewall’s web interface for configuration.

Unfortunately, due to the wide variety of firewalls that may be used, we do not provide specific instructions to cover every type or variation in software or hardware. However, if you have the necessary expertise, we provide abstract firewall rules that can be implemented with iptables, Cisco IOS etc. This guide is based on pfSense, and assumes your firewall hardware has at least three interfaces: WAN, LAN, and OPT1. For hardware, you can build your own network firewall (not covered in this guide) and install pfSense on it. For most installations, we recommend buying a dedicated firewall appliance with pfSense pre-installed, such as the one recommended in the Hardware Guide.

We currently recommend the pfSense SG-3100, which has 3 network interfaces and 6 ports: WAN, OPT1, LAN1, LAN2, LAN3 and LAN4. This firewall comes with an internal switch on the LAN interface. If yours does not you will need to obtain a separate switch to connect the Admin Workstation for the initial installation.

If you are new to pfSense or firewall management in general, we recommend the following resources:

- Official pfSense Wiki
- pfSense: The Definitive Guide
  - Note: This guide is now slightly out of date, although we found it to be a useful reference in the past. To get the latest version of this book, you need to become a pfSense Gold Member.

If you’re using the recommended SG-3100 firewall, then you may find the following resource useful. In particular, you can find instructions on factory resetting the firewall in Chapter 8.

- SG-3100 Product Manual

15.1 Before You Begin

First, consider how the firewall will be connected to the Internet. You will need to provision several unique subnets, which should not conflict with the network configuration on the WAN interface. If you are unsure, consult your local system administrator.
Many firewalls, including the recommended Netgate pfSense, automatically set up the LAN interface on 192.168.1.1/24. This particular private network is also a very common choice for home and office routers. If you are connecting the firewall to a router with the same subnet (common in a small office, home, or testing environment), you will probably be unable to connect to the network at first. However, you will be able to connect from the LAN to the pfSense WebGUI configuration wizard, and from there you will be able to configure the network so it is working correctly.

15.1.1 Configuring Your Firewall

If your firewall has at least 4 NICs, we will refer to the ports as WAN, LAN, OPT1, and OPT2. In this case, we can now use a dedicated port on the network firewall for each component of SecureDrop (Application Server, Monitor Server, and Admin Workstation).

Depending on your network configuration, you should define the following values before continuing.

4 NIC Example

- Admin Subnet: 10.20.1.0/24
- Admin Gateway: 10.20.1.1
- Admin Workstation: 10.20.1.2
- Application Subnet: 10.20.2.0/24
- Application Gateway: 10.20.2.1
- Application Server (OPT1): 10.20.2.2
- Monitor Subnet: 10.20.3.0/24
- Monitor Gateway: 10.20.3.1
- Monitor Server (OPT2): 10.20.3.2

3 NIC Example (SG-3100)

As described earlier, the SG-3100 has an internal switch on the LAN interface which means we can place the Application Server and Admin Workstation on the same subnet and gateway.

- Admin Subnet: 10.20.2.0/24
- Admin Gateway: 10.20.2.1
- Admin Workstation (LAN1): 10.20.2.3
- Application Subnet: 10.20.2.0/24
- Application Gateway: 10.20.2.1
- Application Server (LAN2): 10.20.2.2
- Monitor Subnet: 10.20.3.0/24
- Monitor Gateway: 10.20.3.1
- Monitor Server (OPT1): 10.20.3.2
15.2 Initial Configuration

Unpack the firewall, connect the power, and power on the device.

We will use the pfSense WebGUI to do the initial configuration of the network firewall.¹

15.2.1 Connect to the pfSense WebGUI

1. Boot the Admin Workstation into Tails from the Admin Live USB.

2. Connect the Admin Workstation to the LAN interface. You should see a popup notification in Tails that says “Connection Established”. If you click on the network icon in the upper right of the Tails Desktop, you should see “Wired Connected”:

¹ Tails screenshots were taken on Tails 3.0–beta4. Please make an issue on GitHub if you are using the most recent version of Tails and the interface is different from what you see here.
Warning: Make sure your only active connection is the one you just established with the network firewall. If you are connected to another network at the same time (e.g., a wireless network), you may encounter problems trying to connect the pfSense WebGUI.

3. Launch the Unsafe Browser from the menu bar: Applications Internet Unsafe Browser.

Note: The Unsafe Browser is, as the name suggests, unsafe (its traffic is not routed through Tor). However, it is the only option because Tails intentionally disables LAN access in the Tor Browser.

4. A dialog will ask “Do you really want to launch the Unsafe Browser?” Click Launch.
5. You will see a pop-up notification that says “Starting the Unsafe Browser…”

6. After a few seconds, the Unsafe Browser should launch. The window has a bright red border to remind you to be careful when using it. You should close it once you’re done configuring the firewall and use the Tor Browser for any other web browsing you might do on the Admin Workstation.

7. Navigate to the pfSense WebGUI in the Unsafe Browser: https://192.168.1.1

Note: If you have trouble connecting, go to your network settings and make sure that you have an IPv4 address in the 192.168.1.1/24 range. You may need to turn on DHCP, else you can manually configure a static IPv4
address of 192.168.1.x with a subnet mask of 255.255.255.0. However, make sure not to configure your Tails device to have the same IP as the firewall (192.168.1.1).

8. The firewall uses a self-signed certificate, so you will see a “This Connection Is Untrusted” warning when you connect. This is expected. You can safely continue by clicking Advanced, Add Exception…, and Confirm Security Exception.

9. You should see the login page for the pfSense GUI. Log in with the default username and passphrase (admin / pfsense).
15.2.2 Alternate Hostnames

Before you can set up the hardware firewall, you will need to set the Alternate Hostnames setting after logging in. You will see the Setup Wizard but you should exit out of it by navigating to System -> Advanced. In the Alternate Hostnames dialog box, add 192.168.1.1 as well as the IP address of the Admin Gateway. If you decide against using our recommended defaults for the Admin Gateway, you should include that value here. After saving these settings you should be able to go back to System and select Setup Wizard.

4 NIC Example
3 NIC Example (SG-3100)

Note: If you are using a different IP for the Admin Gateway you should enter that IP in the Alternate Hostname field. Failure to do so will result in an error with the text “An HTTP_REFERER was detected other than what is defined in System -> Advanced”. If you see this error you may have to do a factory reset of the firewall via the serial console.

15.2.3 Setup Wizard

1. If you’re setting up a brand new (or recently factory reset) router, logging in to the pfSense WebGUI will automatically start the Setup Wizard. Click Next, then Next again. Don’t sign up for a pfSense Gold subscription (unless you want to).

2. On the “General Information” page, we recommend leaving your hostname as the default (pfSense). There is no relevant domain for SecureDrop, so we recommend setting this to securedrop.local or something similar. Use your preferred DNS servers. If you don’t know what DNS servers to use, we recommend using Google’s DNS servers: 8.8.8.8 and 8.8.4.4. Click Next.
3. Leave the defaults for “Time Server Information”. Click Next.

4. On “Configure WAN Interface”, enter the appropriate configuration for your network. Consult your local sysadmin if you are unsure what to enter here. For many environments, the default of DHCP will work and the rest of the fields can be left blank. Click Next.

5. a. 4 NIC Example: For “Configure LAN Interface”, use the IP address of the Admin Gateway (10.20.1.1) and the subnet mask (/24) of the Admin Subnet. Click Next.

b. 3 NIC Example (SG-3100): For “Configure LAN Interface”, use the IP address of the Admin Gateway (10.20.2.1) and the subnet mask (/24) of the Admin Subnet. Click Next.
6. Set a strong admin passphrase. We recommend generating a strong passphrase with KeePassX, and saving it in the Tails Persistent folder using the provided KeePassX database template. Click **Next**.

7. Click **Reload**. Once the reload completes and the web page refreshes, click the corresponding “here” link to “continue on to the pfSense webConfigurator”.

At this point, since you (probably) changed the LAN subnet settings from their defaults, you will no longer be able to connect after reloading the firewall and the next request will probably time out. This is not an error - the firewall has reloaded and is working correctly. To connect to the new LAN interface, unplug and reconnect your network cable to get a new network address assigned via DHCP. Note that if you used a subnet with fewer addresses than /24, the default DHCP configuration in pfSense may not work. In this case, you should assign the Admin Workstation a static IP address that is known to be in the subnet to continue.

Now the WebGUI will be available on the Admin Gateway address. Navigate to **https://<Admin Gateway IP>** in the **Unsafe Browser**, and login as before except with the new passphrase you just set for the pfSense WebGUI. Once you've logged in to the WebGUI, you are ready to continue configuring the firewall.

### 15.2.4 Connect Interfaces and Test

Now that the initial configuration is completed, you can connect the WAN port without potentially conflicting with the default LAN settings (as explained earlier). Connect the WAN port to the external network. You can watch the WAN entry in the Interfaces table on the pfSense WebGUI homepage to see as it changes from down (red arrow pointing down) to up (green arrow pointing up). This usually takes several seconds. The WAN’s IP address will be shown once it comes up.

Finally, test connectivity to make sure you are able to connect to the Internet through the WAN. The easiest way to do this is to use **ping** (**Diagnostics → Ping** in the WebGUI). Enter an external hostname or IP that you expect to be up (e.g. google.com) and click “Ping”.
15.3 Disable DHCP on the LAN

pfSense runs a DHCP server on the LAN interface by default. At this stage in the documentation, the Admin Workstation likely has an IP address assigned via that DHCP server.

In order to tighten the firewall rules as much as possible, we recommend disabling the DHCP server and assigning a static IP address to the Admin Workstation instead.

15.3.1 Disable DHCP Server on the Firewall

To disable DHCP, navigate to Services DHCP Server in the pfSense WebGUI. Uncheck the box labeled Enable DHCP server on LAN interface, scroll down, and click the Save button.

15.3.2 Assign a Static IP Address to the Admin Workstation

Now you will need to assign a static IP to the Admin Workstation.

You can easily check your current IP address by clicking the top right of the menu bar, clicking on the Wired Connection and then clicking Wired Settings.
From here you can click on the cog in the lower right of the panel:
This will take you to the network settings, where you can click IPv4 to see whether or not the Automatic (DHCP) or Manual (static IP) setting is turned on.

Change to the IPv4 Settings tab. Change Addresses from Automatic (DHCP) to Manual (if it isn’t already).
Note: The Unsafe Browser will not launch when using a manual network configuration if it does not have DNS servers configured. This is technically unnecessary for our use case because we are only using it to access IP addresses on the LAN, and do not need to resolve anything with DNS. Nonetheless, you should configure some DNS servers here so you can continue to use the Unsafe Browser to access the WebGUI in future sessions.

We recommend keeping it simple and using the same DNS servers that you used for the network firewall in the setup wizard.

4 NIC Example

Fill in the static networking information for the Admin Workstation:

- **Address**: 10.20.1.2
- **Netmask**: 255.255.255.0
- **Gateway**: 10.20.1.1

15.3. Disable DHCP on the LAN
3 NIC Example (SG-3100)

Fill in the static networking information for the Admin Workstation:

- **Address:** 10.20.2.3
- **Netmask:** 255.255.255.0
- **Gateway:** 10.20.2.1
Click **Apply**. If the network does not come up within 15 seconds or so, try disconnecting and reconnecting your network cable to trigger the change. You will need you have succeeded in connecting with your new static IP when you see a pop-up notification that says “Tor is ready. You can now access the Internet”.

**Troubleshooting: DNS Servers and the Unsafe Browser**

After saving the new network configuration, you may still encounter the “No DNS servers configured” error when trying to launch the Unsafe Browser. If you encounter this issue, you can resolve it by disconnecting from the network and then reconnecting, which causes the network configuration to be reloaded.

To do this, click the network icon in the system toolbar, and click **Disconnect** under the name of the currently active network connection, which is displayed in bold. After it disconnects, click the network icon again and click the name of the connection to reconnect. You should see a popup notification that says “Connection Established”, followed several seconds later by the “Tor is ready” popup notification.

For the next step, SecureDrop Configuration, you will manually configure the firewall for SecureDrop, using screen-shots or XML templates as a reference.
15.4 SecureDrop Configuration

SecureDrop uses the firewall to achieve two primary goals:

1. Isolating SecureDrop from the existing network, which may be compromised (especially if it is a venerable network in a large organization like a newsroom).

2. Isolating the Application Server and the Monitor Server from each other as much as possible, to reduce attack surface.

In order to use the firewall to isolate the Application Server and the Monitor Server from each other, we need to connect them to separate interfaces, and then set up firewall rules that allow them to communicate.

15.4.1 Set Up the Firewall Rules

Since there are a variety of firewalls with different configuration interfaces and underlying sets of software, we cannot provide a set of network firewall rules to match every use case.

The easiest way to set up your firewall rules is to look at the screenshots of a correctly configured firewall and edit the interfaces, aliases, and firewall rules on your firewall to match them.

15.4.2 4 NIC Example

If you are using a firewall that has a dedicated interface for each component of SecureDrop, you can follow the below screenshots for setting up your firewall rules.

Set Up OPT1

We set up the LAN interface during the initial configuration. We now need to set up the OPT1 interface for the Application Server. Start by connecting the Application Server to the OPT1 port. Then use the WebGUI to configure the OPT1 interface. Go to Interfaces OPT1, and check the box to Enable Interface. Use these settings:

- IPv4 Configuration Type: Static IPv4
- IPv4 Address: 10.20.2.1 (Application Gateway IP)

Make sure that the CIDR routing prefix is correct (/24). Leave everything else as the default. **Save** and **Apply Changes**.
Set Up OPT2

Next, you will have to enable the OPT2 interface. Go to Interfaces OPT2, and check the box to Enable Interface. OPT2 interface is set up similarly to how we set up OPT1 in the previous section. Use these settings:

- IPv4 Configuration Type: Static IPv4
- IPv4 Address: 10.20.3.1 (Monitor Gateway IP)

Make sure that the CIDR routing prefix is correct (/24). Leave everything else as the default. Save and Apply Changes.
Use Screenshots of Firewall Configuration

Here are some example screenshots of a working pfSense firewall configuration. You will add the firewall rules until they match what is shown on the screenshots.

First, we will configure IP and port aliases. Navigate to Firewall Aliases and you should see a screen with no currently defined IP aliases:

Next you will click Add to add each IP alias. You should leave the Type as Host. Make aliases for the following:

- admin_workstation: 10.20.1.2
- app_server: 10.20.2.2
- external_dns_servers: 8.8.8.8, 8.8.4.4
- monitor_server: 10.20.3.2
- local_servers: app_server, monitor_server
Click **Save** to add the alias.

Keep adding aliases until the screenshot matches what is shown here:
Finally, click **Apply Changes**. This will save your changes. You should see a message “The changes have been applied successfully”: 
Next click “Ports” for the port aliases, and add the following ports:

- OSSEC: 1514
- ossec_agent_auth: 1515

Your configuration should match this screenshot:
Next we will configure firewall rules for each interface. Navigate to **Firewall Rules** to add firewall rules for the LAN, OPT1, and OPT2 interfaces.

**Warning:** Be sure not to delete the Anti-Lockout Rule on the LAN interface. Deleting this rule will lock you out of the pfSense WebGUI.

Add or remove rules until they match the following screenshots by clicking **Add** to add a rule.

**LAN interface:**
OPT1 interface:

<table>
<thead>
<tr>
<th>States</th>
<th>Protocol</th>
<th>Source</th>
<th>Port</th>
<th>Destination</th>
<th>Port</th>
<th>Gateway</th>
<th>Queue</th>
<th>Schedule</th>
<th>Description</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>IPv4</td>
<td>app server</td>
<td>*</td>
<td>monitor server</td>
<td>OSSEC</td>
<td>*</td>
<td>none</td>
<td>OSSEC Agent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>IPv4</td>
<td>app server</td>
<td>*</td>
<td>monitor server</td>
<td>ossec agent auth</td>
<td>*</td>
<td>none</td>
<td>Allow OSSEC agent auth during initial install</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✗</td>
<td>IPv4</td>
<td>OPT1 net</td>
<td>*</td>
<td>LAN net</td>
<td>*</td>
<td>*</td>
<td>none</td>
<td>Block non-whitelisted traffic between OPT1 and LAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✗</td>
<td>IPv4</td>
<td>OPT1 net</td>
<td>*</td>
<td>OPT2 net</td>
<td>*</td>
<td>*</td>
<td>none</td>
<td>Block non-whitelisted traffic between OPT1 and OPT2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>IPv4</td>
<td>app server</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>none</td>
<td>Allow TCP out on any port for Tor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>IPv4</td>
<td>app server</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>none</td>
<td>Allow DNS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>IPv4</td>
<td>app server</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>none</td>
<td>Allow NTP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Finally, click **Apply Changes**. This will save your changes. You should see a message “The changes have been applied successfully”. Once you’ve set up the firewall, exit the Unsafe Browser, and continue with the “Keeping pfSense up to date” section below.

### 15.4.3 3 NIC Example (SG-3100)

The below guide assumes you are using a 3 NIC firewall such as the SG-3100. While the SG-3100 has an integrated switch, you may need to add a switch to the LAN interface if you use a different firewall.

**Set Up LAN**

Although we set up the LAN interface during the Setup Wizard we need to make a few revisions. Navigate in the WebGUI to configure the LAN interface. Go to **Interfaces LAN**, and ensure the **Enable Interface** box is checked. Use these settings:

- IPv4 Configuration Type: Static IPv4
- IPv4 Address: 10.20.2.1 (Application Gateway IP)

Make sure that the CIDR routing prefix is correct (/24). Leave everything else as the default. **Save** and **Apply Changes**.
Disable Anti-Lockout Rule

In order to further lockdown communication we will disable the rule that allows traffic to the firewall over the local network. Navigate to System Advanced and find the “Anti-lockout” rule. Ensure the box is checked and save the configuration.
Warning: Do not reboot the firewall until after you have set the firewall rules based on the screenshots below. If you get locked out of the firewall you may have to factory reset it by connecting to it over the serial console.

Set Up OPT1

Next, you will have to enable the OPT1 interface. Go to Interfaces OPT1, and check the box to Enable Interface. Use these settings:

- IPv4 Configuration Type: Static IPv4
- IPv4 Address: 10.20.3.1 (Monitor Gateway IP)

Make sure that the CIDR routing prefix is correct (/24). Leave everything else as the default. Save and Apply Changes.
Use Screenshots of Firewall Configuration

Here are some example screenshots of a working pfSense firewall configuration. You will add the firewall rules until they match what is shown on the screenshots.

First, we will configure IP and port aliases. Navigate to **Firewall Aliases** and you should see a screen with no currently defined IP aliases:

Next you will click **Add** to add each IP alias. You should leave the **Type** as **Host**. Make aliases for the following:

- admin_workstation: 10.20.2.3
- app_server: 10.20.2.2
- **external_dns_servers**: 8.8.8.8, 8.8.4.4
- **monitor_server**: 10.20.3.2
- **local_servers**: app_server, monitor_server

Click **Save** to add the alias.

Keep adding aliases until the screenshot matches what is shown here:
Finally, click **Apply Changes**. This will save your changes. You should see a message “The changes have been applied successfully”:
Next click “Ports” for the port aliases, and add the following ports:

- **OSSEC**: 1514
- **ossec_agent_auth**: 1515

Your configuration should match this screenshot:
Next we will configure firewall rules for each interface. Navigate to **Firewall Rules** to add firewall rules for the LAN and OPT1 interfaces.

**Warning:** Be sure not to delete the Anti-Lockout Rule on the LAN interface. Deleting this rule will lock you out of the pfSense WebGUI.

Add or remove rules until they match the following screenshots by clicking **Add** to add a rule.

**LAN interface:**
Finally, click **Apply Changes**. This will save your changes. You should see a message “The changes have been applied successfully”. Once you’ve set up the firewall, exit the Unsafe Browser, and continue with the “Keeping pfSense up to date” section below.
15.4.4 Configuration Reference Templates

As an alternative to the provided screenshots, you can examine the provided .xml templates as a reference:

- **Interfaces config**: `install_files/network_firewall/interfaces-config-pfSense.xml`
- **Aliases**: `install_files/network_firewall/aliases-config-pfSense.xml`
- **Firewall rules**: `install_files/network_firewall/filter-config-pfSense.xml`

**Note**: These will not load using pfSense Restore and are here as a reference only. See [GitHub #2282](https://github.com/pfSense/pfsense/issues/2282) for more info.

15.5 Tips for Setting Up pfSense Firewall Rules

Here are some general tips for setting up pfSense firewall rules:

1. Create aliases for the repeated values (IPs and ports).

2. pfSense is a stateful firewall, which means that you don’t need corresponding rules to allow incoming traffic in response to outgoing traffic (like you would in, e.g. `iptables --state ESTABLISHED,RELATED`). pfSense does this for you automatically.

3. You should create the rules on the interface where the traffic originates.

4. Make sure you delete the default “allow all” rule on the LAN interface. Leave the “Anti-Lockout” rule enabled.

5. Any traffic that is not explicitly passed is logged and dropped by default in pfSense, so you don’t need to add explicit rules (`iptables LOGNDROP`) for that.

6. Since some of the rules are almost identical except for whether they allow traffic from the Application Server or the Monitor Server, you can use the “add a new rule based on this one” button to save time creating a copy of the rule on the other interface.

7. If you are troubleshooting connectivity, the firewall logs can be very helpful. You can find them in the WebGUI in `Status → System Logs → Firewall`.

15.6 Keeping pfSense up to Date

Periodically, the pfSense project maintainers release an update to the pfSense software running on your firewall. You will be notified by the appearance of text saying that there is a new version in the **Version** section of the “Status: Dashboard” page (the home page of the WebGUI).
If you see that an update is available, we recommend installing it. Most of these updates are for minor bugfixes, but occasionally they can contain important security fixes. You should keep apprised of updates yourself by checking the pfSense Blog posts with the “releases” tag.

**Note:** Protip: Subscribe to the RSS feed.

To install the update, click the Download icon next to the update then click the “Confirm” button:
You will see a page with a progress bar while pfSense performs the upgrade:

<table>
<thead>
<tr>
<th>Branch</th>
<th>Latest stable version (2.4.x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Base System</td>
<td>2.4.4.2</td>
</tr>
<tr>
<td>Latest Base System</td>
<td>2.4.4.3</td>
</tr>
</tbody>
</table>
Note: This may take a while, so be patient!

Once it is complete, you will see a notification of successful upgrade:
15.7 Abstract Firewall Rules

The pfSense instructions using the web interface can also be precisely described as follows:

- Disable DHCP (in case the firewall is providing a DHCP server by default)
- Disallow all traffic by default (inbound or outbound)
- Allow UDP OSSEC (port 1514) from Application Server to Monitor Server
- Allow TCP ossec agent auth (port 1515) from Application Server to Monitor Server
- Allow TCP/UDP DNS from Application Server and Monitor Server to the IPs of known name servers
- Allow UDP NTP from Application Server and Monitor Server to all
- Allow TCP any port from Application Server and Monitor Server to all (this is needed for making connections to the Tor network)
- Allow TCP 80/443 from Admin Workstation to all (in case there is a need to access the web interface of the firewall)
- Allow TCP ssh from Admin Workstation to Application Server and Monitor Server
- Allow TCP any port from Admin Workstation to all
This can be implemented with iptables, Cisco IOS etc. if you have the necessary expertise.
16.1 Install Ubuntu

**Caution:** Please ensure you are using Ubuntu Xenial ISO images 16.04.6 or greater. Ubuntu Xenial ISO images 16.04.5 and lower ship with a version of the `apt` package vulnerable to CVE-2019-3462. If you are using 16.04.5 or lower, the initial base OS must be installed without Internet connectivity.

**Note:** Installing Ubuntu is simple and may even be something you are very familiar with, but we **strongly** encourage you to read and follow this documentation exactly as there are some “gotchas” that may cause your SecureDrop set up to break.

The SecureDrop Application Server and Monitor Server run **Ubuntu Server 16.04.6 LTS (Xenial Xerus)**. To install Ubuntu on the servers, you must first download and verify the Ubuntu installation media. You should use the **Admin Workstation** to download and verify the Ubuntu installation media.

### 16.1.1 Download the Ubuntu Installation Media

The installation media and the files required to verify it are available on the Ubuntu Releases page. You will need to download the following files:

- `ubuntu-16.04.6-server-amd64.iso`
- `SHA256SUMS`
- `SHA256SUMS.gpg`

If you’re reading this documentation in the Tor Browser on the **Admin Workstation**, you can just click the links above and follow the prompts to save them to your Admin Workstation. We recommend saving them to the `/home/amnesia/Persistent/Tor Browser` directory on the **Admin Workstation**, because it can be useful to have a copy of the installation media readily available.
16.1.2 Verify the Ubuntu Installation Media

You should verify the Ubuntu image you downloaded hasn’t been modified by a malicious attacker or otherwise corrupted. We can do so by checking its integrity with cryptographic signatures and hashes.

First, we will download Ubuntu Image Signing Key and verify its fingerprint.

```
gpg --recv-key "C598 6B4F 1257 FFA8 6632 CBA7 4618 1433 FBB7 5451"
```

**Note:** It is important you type this out correctly. If you are not copy-pasting this command, we recommend you double-check you have entered it correctly before pressing enter.

Again, when passing the full public key fingerprint to the `--recv-key` command, GPG will implicitly verify that the fingerprint of the key received matches the argument passed.

**Caution:** If GPG warns you that the fingerprint of the key received does not match the one requested do not proceed with the installation. If this happens, please email us at securedrop@freedom.press.

Verify the SHA256SUMS file and move on to the next step if you see “Good Signature” in the output.

```
gpg --verify SHA256SUMS.gpg SHA256SUMS
```

The next and final step is to verify the Ubuntu image.

```
sha256sum -c <(grep ubuntu-16.04.6-server-amd64.iso SHA256SUMS)
```

If the final verification step is successful, you should see the following output in your terminal.

```
ubuntu-16.04.6-server-amd64.iso: OK
```

**Caution:** If you do not see the line above it is not safe to proceed with the installation. If this happens, please contact us at securedrop@freedom.press.

16.1.3 Create the Ubuntu Installation Media

To create the Ubuntu installation media, you can either burn the ISO image to a CD-R or create a bootable USB stick. As a reliable method we recommend using the `dd` command to copy the hybrid ISO directly to a USB drive rather than a utility like UNetbootin which can result in errors. Once you have a CD or USB with an ISO image of Ubuntu on it, you may begin the Ubuntu installation on both SecureDrop servers.
To use `dd` you first need to find where the USB device you wish to install Ubuntu on has been mapped. Simply running the command `lsblk` in the terminal will give you a list of your block storage device mappings (this includes hard drives and USB). If the USB you are writing the Ubuntu installer to is of a different size or brand than the USB you are running Tails from, it should be easy to identify which USB has which sdX identifier. If you are unsure, try running `lsblk` before and after plugging in the USB you are using for the Ubuntu installer.

If your USB is mapped to `/dev/sdX` and you are currently in the directory that contains the Ubuntu ISO, you would use `dd` like so:

```bash
sudo dd conv=fdatasync if=ubuntu-16.04.6-server-amd64.iso of=/dev/sdX
```

## 16.1.4 Perform the Installation

The steps below are the same for both the *Application Server* and the *Monitor Server*.

Start by inserting the Ubuntu installation media into the server. Boot or reboot the server with the installation media inserted, and enter the boot menu. To enter the boot menu, you need to press a key as soon as you turn the server on. This key varies depending on server model, but common choices are Esc, F2, F10, and F12. Often, the server will briefly display a message on boot that shows which key should be pressed to enter the boot menu. Once you’ve entered the boot menu, select the installation media (USB or CD) and press Enter to boot it.

After booting the Ubuntu image, select **Install Ubuntu Server**.

Follow the steps to select your language, country and keyboard settings. Once that’s done, let the installation process continue.
16.1.5 Configure the Network Manually

The Ubuntu installer will try to autoconfigure networking for the server you are setting up; however, SecureDrop requires manual network configuration. You can hit Cancel at any point during network autoconfiguration to be given the choice to Configure the network manually.

If network autoconfiguration completes before you can do this, the next window will ask for your hostname. To get back to the choice of configuring the network manually, Cancel the step that asks you to set a hostname and choose the menu option that says Configure the network manually instead.

For a production install with a pfSense network firewall in place, the Application Server and the Monitor Server are on separate networks. You may choose your own network settings at this point, but make sure the settings you choose are unique on the firewall’s network and remember to propagate your choices through the rest of the installation process.

Below are the configurations you should enter, assuming you used the network settings from the network firewall guide for a 3 NIC or 4 NIC firewall. If you did not, adjust these settings accordingly.

- **Application Server**:
  - Server IP address: 10.20.2.2
  - Netmask (default is fine): 255.255.255.0
  - Gateway: 10.20.2.1
  - For DNS, use Google’s name servers: 8.8.8.8 and 8.8.4.4
  - Hostname: app
  - Domain name should be left blank

- **Monitor Server**:
  - Server IP address: 10.20.3.2
  - Netmask (default is fine): 255.255.255.0
  - Gateway: 10.20.3.1
  - For DNS, use Google’s name servers: 8.8.8.8 and 8.8.4.4
  - Hostname: mon
  - Domain name should be left blank

16.1.6 Continue the Installation

You can choose whatever username and passphrase you would like. To make things easier later you should use the same username and same passphrase on both servers (but not the same passphrase as username). Make sure to save this passphrase in your admin KeePassX database afterwards.

Click ‘no’ when asked to encrypt the home directory. Then configure your time zone.

16.1.7 Partition the Disks

Before setting up the server’s disk partitions and filesystems in the next step, you will need to decide if you would like to enable Full Disk Encryption (FDE). If the servers are ever powered down, FDE will ensure all of the information on them stays private in case they are seized or stolen.
Warning: The Ansible playbooks for SecureDrop will enable nightly reboots after the `cron-apt` task runs for automatic updates. Using FDE would therefore require manual intervention every morning. Consequently we strongly discourage the use of FDE.

While FDE can be useful in some cases, we currently do not recommend that you enable it because there are not many scenarios where it will be a net security benefit for SecureDrop operators. Doing so will introduce the need for more passphrases and add even more responsibility on the admin of the system (see this GitHub issue for more information).

If you wish to proceed without FDE as recommended, choose the installation option that says `Guided - use entire disk and set up LVM`.

However, if you decide to go ahead and enable FDE, please note that doing so means SecureDrop will become unreachable after an automatic reboot. An admin will need to be on hand to enter the passphrase in order to decrypt the disks and complete the startup process, which will occur anytime there is an automatic software update, and also several times during SecureDrop’s installation. We recommend that the servers be integrated with a monitoring solution that so that you receive an alert when the system becomes unavailable.

To enable FDE, select `Guided - use entire disk and set up encrypted LVM` during the disk partitioning step and write the changes to disk. Follow the recommendations as to choosing a strong passphrase. As the admin, you will be responsible for keeping this passphrase safe. Write it down somewhere and memorize it if you can. If inadvertently lost it could result in total loss of the SecureDrop system.

After selecting either of those options you may be asked a few questions about overwriting anything currently on the server you are using. Select yes. You do not need an HTTP proxy, so when asked, you can just click continue.

16.1.8 Finish the Installation

Wait for the base system to finish installing. When you get to the `Configure tasksel` screen, choose `No automatic updates`. The subsequent SecureDrop installation will include a task that handles regular software updates.

Note: The Ansible playbooks for SecureDrop will configure automatic updates via `cron-apt`. As part of the automatic update process, the servers will reboot nightly. See the `OSSEC guide` for example notifications generated by the reboots.

When you get to the software selection screen, deselect the preselected `Standard system utilities` and select `OpenSSH server` by highlighting each option and pressing the space bar.

Caution: Hitting enter before the space bar will force you to start the installation process over.

Once `OpenSSH Server` is selected, hit `Continue`.

You will then have to wait for the packages to finish installing.

When the packages are finished installing, Ubuntu will automatically install the bootloader (GRUB). If it asks to install the bootloader to the Master Boot Record, choose `Yes`. When everything is done, reboot.

16.1.9 Save the Configurations

When you are done, make sure you save the following information:

- The IP address of the `Application Server`
16.2 Test Connectivity

Now that the firewall is set up, you can plug the Application Server and the Monitor Server into the firewall. If you are using a setup where there is a switch on the LAN port, plug the Application Server into the switch and plug the Monitor Server into the OPT1 port.

You should make sure you can connect from the Admin Workstation to both of the servers before continuing with the installation.

In a terminal, verify that you can SSH into both servers, authenticating with your passphrase:

```
$ ssh <username>@<App IP address> hostname
app
$ ssh <username>@<Monitor IP address> hostname
mon
```

Tip: If you cannot connect, check the network firewall logs for clues.

16.3 Set Up SSH Keys

Ubuntu’s default SSH configuration authenticates users with their passphrases; however, public key authentication is more secure, and once it’s set up it is also easier to use. In this section, we will create a new SSH key for authenticating to both servers. Since the Admin Live USB was set up with SSH Client Persistence, this key will be saved on the Admin Live USB and can be used in the future to authenticate to the servers in order to perform administrative tasks.

First, generate the new SSH keypair:

```
ssh-keygen -t rsa -b 4096
```

You’ll be asked to “Enter file in which to save the key” Type Enter to use the default location.

Given that this key is on the encrypted persistence of a Tails USB, you do not need to add an additional passphrase to protect the key. If you do elect to use a passphrase, note that you will need to manually type it (Tails’ pinentry will not allow you to copy and paste a passphrase).

Once the key has finished generating, you need to copy the public key to both servers. Use ssh-copy-id to copy the public key to each server, authenticating with your passphrase:

```
ssh-copy-id <username>@<App IP address>
ssh-copy-id <username>@<Mon IP address>
```

Verify that you are able to authenticate to both servers by running the below commands. You should not be prompted for a passphrase (unless you chose to passphrase-protect the key you just created).

```
$ ssh <username>@<App IP address> hostname
app
$ ssh <username>@<Monitor IP address> hostname
mon
```
If you have successfully connected to the server via SSH, the terminal output will be name of the server to which you have connected (‘app’ or ‘mon’) as shown above.
17 Install SecureDrop

17.1 Install Prerequisites

SecureDrop has dependencies that need to be loaded onto the Admin Workstation before installing the servers. To install these dependencies, from the base of the SecureDrop repository (~/Persistent/securedrop/) run the following command:

```
./securedrop-admin setup
```

The package installation will take approximately 10 minutes or longer, depending on network speed and computing power.

**Note:** On Tails 3.9 or later, the apt persistence feature will prompt to install the package automatically from persistent storage on each boot. These apt packages don’t need to persist, click on **Install Only Once**:

```
Add htop to your additional software?
To install it automatically from your persistent storage when starting Tails.

Install Only Once  Install Every Time
```

**Note:** Occasionally this command times out due to network latency issues. You should be able to re-run the command and complete the setup. If you run into a problem, try removing the ~/Persistent/securedrop/admin/.venv/ directory and the ~/Persistent/securedrop/.venv symbolic link and running the command again.
Important: The setup command should only be run as the amnesia user, not as root. Contact the SecureDrop team if the package installation encounters repeated errors.

17.2 Localization of the Source Interface and Journalist Interface

The Source Interface and Journalist Interface are translated in the following languages:

- Arabic (ar)
- Catalan (ca)
- German (de_DE)
- Greek (el)
- Spanish (es_ES)
- French (fr_FR)
- Hindi (hi)
- Icelandic (is)
- Italian (it_IT)
- Norwegian (nb_NO)
- Dutch (nl)
- Portuguese, Brasil (pt_BR)
- Romanian (ro)
- Russian (ru)
- Swedish (sv)
- Turkish (tr)
- Chinese, Traditional (zh_Hant)

During the installation you will be given the opportunity to choose from a list of supported languages to display using the codes shown in parentheses.

Note: With a Source Interface displayed in French (for example), sources submitting documents are likely to expect a journalist fluent in French to be available to read the documents and follow up in that language.

17.3 Onion Service Options

SecureDrop supports the use of traditional (v2) or next-generation (v3) onion services for the Source and Journalist Interfaces, as well as the SSH proxy services if they are configured. Either or both may be enabled, but we recommend the use of v3 onion services for any new instances, as they offer greater security.

For more information on v3 onion services, including upgrade options for existing instances, see SecureDrop v3 onion services.
17.4 Configure the Installation

Make sure you have the following information and files ready before continuing:

- the Application Server local IP address
- the Monitor Server local IP address
- the Submission Public Key (from the Transfer Device)
- the Submission Key fingerprint
- the email address that will receive alerts from OSSEC
- the GPG public key and fingerprint for the email address that will receive the alerts
- connection information for the SMTP relay that handles OSSEC alerts (see the OSSEC Alerts Guide)
- the username of a journalist who will be using SecureDrop (you can add more later)
- the username of the system admin

Optionally, you can configure daily email notifications of submission activity for journalists. These help journalists avoid spending time checking the Journalist Interface when there are no submissions. For this you will need:

- the journalist alerts GPG key
- the journalist alerts GPG key fingerprint
- the email address that will receive the journalist alerts

**Note:** It is not possible to specify multiple email addresses for email notifications. If there are multiple intended recipients, use an alias or mailing list. However, all subscribers must share the GPG private key, as it is not possible to specify multiple keys.

**Note:** The journalist notification is sent after the daily reboot of the Application Server.

Before proceeding, you will need to copy the following files to `install_files/ansible-base`:

- the Submission Public Key file
- the admin’s GPG public key file (for encrypting OSSEC alerts)

The Submission Public Key should be located on your Transfer Device from earlier. Its exact path will depend on the location where the USB stick is mounted. From the root of the SecureDrop repository, run:

```
cp /media/[USB folder]/SecureDrop.asc install_files/ansible-base
```

where `/media/[USB folder]/` corresponds to the Transfer Device. (You can also use the copy and paste capabilities of the file manager.)

Repeat this step for the Admin GPG key.

Next, run the configuration playbook and answer the prompts with values that match your environment:

```
./securedrop-admin sdconfig
```

The script will automatically validate the answers you provided and display error messages if any problems are detected. The answers will be written to the file `install_files/ansible-base/group_vars/all/site-specific`.  

17.4. Configure the Installation 147
When you’re done, save the file and quit the editor.

## 17.5 Install SecureDrop Servers

Now you are ready to install! This process will configure the servers and install SecureDrop and all of its dependencies on the remote servers.

```
./securedrop-admin install
```

You will be prompted to enter the `sudo` passphrase for the **Application Server** and **Monitor Server** (which should be the same).

The installation process will take some time. It will return you to the terminal prompt when complete.

If any errors occur while running the install, carefully inspect the error output. Considering saving any error messages for reference and troubleshooting.

**Note:** If you see an error running `./securedrop-admin install`, and believe it may be an intermittent issue (for example, due to losing network connectivity to the servers), it is safe to run the `./securedrop-admin install` command again. If you see the same issue consistently, then you will need to troubleshoot it.

If you see the error message “timeout (62s) waiting for privilege escalation prompt”, try deleting the Ansible control path directory on your **Admin Workstation** (`rm -rf ~/.ansible/cp`) to reset the connection to the servers, then re-run the `./securedrop-admin install` command from within `~/Persistent/securedrop`.

If you encounter other errors, we encourage you to submit a bug report, or to contact us at securedrop@freedom.press (GPG encrypted).

If needed, make edits to the file located at `install_files/ansible-base/group_vars/all/site-specific` as described above. If you continue to have issues, please submit a detailed issue notice on GitHub or send an email to securedrop@freedom.press.

**Note:** The SecureDrop install process configures a custom Linux kernel hardened with the grsecurity patch set. Only binary images are hosted in the apt repo. For source packages, see the *Source Offer*.

Once the installation is complete, addresses and credentials for each onion service will be available in the following files under `install_files/ansible-base`:

### 17.5.1 V2 onion services

- `app-source-ths` contains the `.onion address of the Source Interface.`
- `app-journalist-aths` contains the HidServAuth configuration line for the **Journalist Interface**. During a later step, this will be automatically added to your Tor configuration file in order to exclusively limit connections to the hidden service.
- `app-ssh-aths` contains the HidServAuth for SSH access to the **Application Server**.
- `mon-ssh-aths` contains the HidServAuth for SSH access to the **Monitor Server**.
Warning: The `app-journalist-aths`, `app-ssh-aths`, and `mon-ssh-aths` files contain passwords for their corresponding authenticated onion services. They should not be shared with third parties or copied from the `Admin Workstation` for any reason other than well-defined administrative tasks such as onboarding new users or performing backups.

If v3 onion services are not enabled, the dynamic inventory file will automatically read the Onion URLs from the `app-ssh-aths` and `mon-ssh-aths` files and use them to connect to the servers over SSH during subsequent playbook runs.

### 17.5.2 V3 onion services

- `app-sourcev3-ths` contains the v3 `.onion` address of the **Source Interface**.
- `app-journalist.auth_private` contains the onion address and private key providing access to the **Journalist Interface**.
- `app-ssh.auth_private` contains the onion address and private key providing SSH access to the **Application Server**.
- `mon-ssh.auth_private` contains the onion address and private key providing SSH access to the **Monitor Server**.
- `tor_v3_keys.json` contains the keypairs required for access to the **Journalist Interface** and SSH access to the servers - it is required for future runs of `./securedrop-admin install`.

Warning: The three `.auth_private` files and the `tor_v3_keys.json` file contain secret keys that should not be shared with third parties, or copied from the `Admin Workstation` for any purpose other than tasks such as performing backups or onboarding new users.

The dynamic inventory file will automatically read the onion addresses from the `app-ssh.auth_private` and `mon-ssh.auth_private` files and use them to connect to the servers over SSH during subsequent playbook runs.
18.1 Auto-connect to the Authenticated Tor Onion Services

The SecureDrop installation process adds multiple layers of authentication to protect access to the most sensitive assets in the SecureDrop system:

1. The Journalist Interface, because it provides access to submissions (although they are encrypted to an offline key), and some metadata about sources and submissions.
2. SSH on the Application Server
3. SSH on the Monitor Server

The installation process blocks direct access to each of these assets, and sets up Authenticated Tor Onion Services (ATHS) to provide authenticated access instead. Authenticated Tor Onion Services share the benefits of Tor Hidden Services, but are only accessible to users who possess a shared secret (auth-cookie in the Tor documentation) that is generated during the hidden service setup process.

In order to access an ATHS, you need to add one or more “auth-cookie” values to your Tor configuration file (torrc) and restart Tor. Doing this manually is annoying and error-prone, so SecureDrop includes a set of scripts in ./tails_files that can set up a Tails instance to automatically configure Tor to access a set of ATHS. In order to persist these changes across reboots, the Tails instance must have persistence enabled (specifically, the “dotfiles persistence”).

Note: Starting in version 0.3.7, SecureDrop requires Tails 2.x or greater.

To install the auto-connect configuration, start by navigating to the directory with these scripts (~/Persistent/securedrop/), and run the install script:

```
./securedrop-admin tailsconfig
```

Type the Administration Password that you selected when starting Tails and hit Enter. This script installs a persistent script that runs every time you connect to a network in Tails, and automatically configures access to the Journalist Interface and to the servers via SSH. The HidServAuth info is collected from
files in ~/Persistent/securedrop/install_files/ansible-base and stored in ~/Persistent/.securedrop/torrc_additions thereafter.

**Tip:** Copy the files app-journalist-aths and app-source-ths to the Transfer Device in preparation for setting up the Journalist Workstation. Then you can use the securedrop-admin tool to configure access for Journalists as well.

In addition, the script creates desktop and menu shortcuts for the Source and Journalist Interfaces, directs Tails to install Ansible at the beginning of every session, and sets up SSH host aliases for the servers.

The only thing you need to remember to do is enable persistence when you boot the Admin Workstation. If you are using the Admin Workstation and are unable to connect to any of the authenticated Onion Services, restart Tails and make sure to enable persistence.

### 18.2 Back Up the Workstations

USB drives can wear out, get lost, or otherwise become corrupted, making it very important to be sure to keep current backups. Follow the Back up the Workstations document to create a backup of your Secure Viewing Station, Admin Workstation, and Journalist Workstations after you’ve completed the installation and post-installation steps.
Create an Admin Account on the Journalist Interface

In order for any user (admin or journalist) to access the Journalist Interface, they need:

1. The auth-cookie for the Journalist Interface’s ATHS
2. An account on the Journalist Interface, which requires the following credentials to log in:
   - Username
   - Passphrase
   - Two-factor authentication code

You should create a separate account on the Journalist Interface for each user who needs access. This makes it easy to enable or disable access to the Journalist Interface on an individual basis, so you can grant access to new users or revoke access for users who have left the organization or should no longer be allowed to access the Journalist Interface.

There are two types of accounts on the Journalist Interface: admin accounts and normal accounts. Admins accounts are like normal accounts, but they are additionally allowed to manage (add, change, delete) other user accounts through the web interface.

You must create the first admin account on the Journalist Interface by running a command on the Application Server. After that, the Journalist Interface admin can create additional accounts through the web interface.

If you are installing SecureDrop yourself, to create the first admin account, SSH to the Application Server, then:

```bash
sudo -u www-data bash
cd /var/www/securedrop
./manage.py add-admin
```

Follow the prompts.

A secure diceare passphrase will be generated by manage.py. You will see output like this:

```
This journalist’s passphrase is: delivery propose requisite stunner dragonfly — unstamped stowaway
```

Passphrases include the spaces between the words, but not leading or trailing whitespace. Be sure to save this passphrase in the appropriate KeePassX database.
Once that’s done, you should open the Tor Browser and navigate to the *Journalist Interface*’s .onion address. Verify that you can log in to the *Journalist Interface* with the admin account you just created.

For adding more user accounts, please refer now to our *Admin Interface Guide*.

**Note:** You can now set a custom logo image on your web interfaces by following the *Updating Logo Image* documentation.
CHAPTER 20

Test the Installation

20.1 Test Connectivity

20.1.1 SSH to Both Servers Over Tor

Assuming you haven’t disabled ssh over tor, SSH access will be restricted to the tor network.

On the Admin Workstation, you should be able to SSH to the Application Server and the Monitor Server.

```
ssh app
ssh mon
```

The SSH aliases should have been configured automatically by running the `./securedrop-admin tailsconfig` tool. If you’re unable to connect via aliases, try using the verbose command format to troubleshoot:

```
ssh <username>@<app.onion>
ssh <username>@<mon.onion>
```

**Tip:** If your instance uses v2 onion services, you can find the Onion URLs for SSH in `app-ssh-aths` and `mon-ssh-aths` inside the `install_files/ansible-base` directory. If your instance uses v3 onion services, check the `app-ssh.auth_private` and `mon-ssh.auth_private` files instead.

20.1.2 Log in to Both Servers via TTY

All access to the SecureDrop servers should be performed over SSH from the Admin Workstation. To aid in troubleshooting, login via a physical keyboard attached to the server is also supported.
20.2 Sanity-Check the Installation

On each server:

1. Check that you can execute privileged commands by running `sudo su`.
2. Verify that you are booted into a grsec kernel: run `uname -r` and verify that the name of the running kernel ends with `-grsec`.
3. Check the current applied iptables rules with `iptables-save`. It should output approximately 50 lines.
4. You should have received an email alert from OSSEC when it first started. If not, review our OSSEC Alerts Guide.

On the Application Server:

1. Check the AppArmor status with `sudo aa-status`. On a production instance all profiles should be in enforce mode.

20.3 Test the Web Interfaces

1. Make sure the Source Interface is available, and that you can make a submission.
   - Open the Source Interface in the Tor Browser by clicking on its desktop shortcut. Proceed through the codename generation (copy this down somewhere) and submit a test message or file.
   - Usage of the Source Interface is covered by our Source User Manual.
2. Test that you can access the Journalist Interface, and that you can log in as the admin user you just created.
   - Open the Journalist Interface in the Tor Browser by clicking on its desktop shortcut. Enter your passphrase and two-factor authentication code to log in.
   - If you have problems logging in to the Admin/Journalist Interface, SSH to the Application Server and restart the ntp daemon to synchronize the time: `sudo service ntp restart`. Also check that your smartphone’s time is accurate and set to network time in its device settings.
3. Test replying to the test submission.
   - While logged in as an admin, you can send a reply to the test source submission you made earlier.
   - Usage of the Journalist Interface is covered by our Journalist User Manual.
4. Test that the source received the reply.
   - Within Tor Browser, navigate back to the Source Interface and use your previous test source codename to log in (or reload the page if it’s still open) and check that the reply you just made is present.
5. Remove the test submissions you made prior to putting SecureDrop to real use. On the main Journalist Interface page, select all sources and click Delete selected.

Once you’ve tested the installation and verified that everything is working, see How to Use SecureDrop.
Congratulations! You’ve successfully installed SecureDrop.

At this point, the only person who has access to the system is the admin. In order to grant access to journalists, you will need to do some additional setup for each individual journalist.

In order to use SecureDrop, each journalist needs two things:

1. A **Journalist Tails USB**.
   
The *Journalist Interface* is only accessible as an authenticated onion service. For ease of configuration and security, we require journalists to set up a Tails USB with persistence that they are required to use to access the *Journalist Interface*.

2. Access to the **Secure Viewing Station**.
   
The *Journalist Interface* allows journalists to download submissions from sources, but they are encrypted to the offline private key that is stored on the Secure Viewing Station Tails USB. In order for the journalist to decrypt and view submissions, they need access to a *Secure Viewing Station*.

### 21.1 Determine Access Protocol for the **Secure Viewing Station**

Currently, SecureDrop only supports encrypting submissions to a single public/private key pair - the *Submission Key*. As a result, each journalist needs a way to access the *Secure Viewing Station* with a Tails USB that includes the *Submission Private Key*.

The access protocol for the *Secure Viewing Station* depends on the structure and distribution of your organization. If your organization is centralized and there are only a few journalists with access to SecureDrop, they should be fine with sharing a single Secure Viewing Station. On the other hand, if your organization is distributed, or if you have a lot of journalists who wish to access SecureDrop concurrently, you will need to provision multiple *Secure Viewing Stations*. 
21.2 Create a Journalist Tails USB

Each journalist will need a Journalist Tails USB and a Journalist Workstation, which is the computer they use to boot their Tails USB.

To create a Journalist Interface Tails USB, just follow the same procedure you used to create a Tails USB with persistence for the Admin Workstation, as documented in the Tails Setup Guide.

Note: As with your Admin Workstation, you can use a fresh copy of the blank KeePassX template in the repository to initialize the password database on the Journalist Workstation. You can safely edit this copy to remove sections or fields that are not relevant for the journalist you are onboarding. For example, the admin section of the password database should never be filled in on a Journalist Workstation.

Once you’re done, boot into the new Journalist Tails USB on the Journalist Workstation. Enable persistence and set an admin passphrase before continuing with the next section.

21.3 Set Up Automatic Access to the Journalist Interface

Since the Journalist Interface is an authenticated onion service, you must set up the Journalist Workstation to auto-configure Tor, similarly to the Admin Workstation. The procedure is essentially identical, except the SSH configuration will be skipped, since only admins need to access the servers over SSH.

• First, boot into the Admin Workstation. If your instance has not been set up to use v3 onion services, copy the following v2 service files to a Transfer Device:

```
~/Persistent/securedrop/install_files/ansible_base/app-source-ths
~/Persistent/securedrop/install_files/ansible_base/app-journalist-aths
```

If your instance was set up to use v3 services, copy the following files instead:

```
~/Persistent/securedrop/install_files/ansible_base/app-sourcev3-ths
~/Persistent/securedrop/install_files/ansible_base/app-journalist.auth_private
```

Then, boot into the new Journalist Workstation USB.

Warning: Do not copy the app-ssh-aths, mon-ssh-aths, app-ssh.auth_private, mon-ssh. auth_private, or tor_v3_keys.json files to the Journalist Workstation. Those files contain private keys and authentication information for SSH server access. Only the Admin Workstation should have shell access to the servers.

• Install the SecureDrop application code on the workstation’s persistent volume, following the documentation for cloning the SecureDrop repository.

• Copy the files from the Transfer Device to ~/Persistent/securedrop/install_files/ansible-base

• Open a terminal and run the following commands:

```
cd ~/Persistent/securedrop
./securedrop-admin setup
./securedrop-admin tailsconfig
```
Note: The setup command may take several minutes, and may fail partway due to network issues. If so, run it again before proceeding.

- Once the tailsconfig command is complete, verify that the Source and Journalist Interfaces are accessible at their v2 addresses via the SecureDrop desktop shortcuts.
- Securely wipe the files on the Transfer Device, by right-clicking them in the file manager and selecting Wipe.

Warning: The app-journalist-aths and app-journalist.auth_private files contain secret authentication information for the authenticated onion service used by the Journalist Interface, and should not be shared except through the onboarding process.

21.4 Add an account on the Journalist Interface

Finally, you need to add an account on the Journalist Interface so the journalist can log in and access submissions. See the section on Adding Users in the admin Guide.

21.5 Provision a personal Transfer Device and Export Device

In small organizations, a team of journalists may want to share a single Transfer Device and a single Export Device. In larger organizations, you may want to provision a personal Transfer Device and Export Device for each journalist who may need to copy files off the Secure Viewing Station. Please see the setup guide for more information.

21.6 Verify Journalist Setup

Once the journalist device and account have been provisioned, then the admin should run through the following steps with each journalist to verify the journalist is set up for SecureDrop.

The journalist should verify that they:

1. Have their own Journalist Workstation USB drive that they are able to boot on the computer designated for this purpose (which can be their everyday laptop).

Note: It is important that they test exactly on the computer they will be using as the Journalist Workstation, as there can be differences in Tails compatibility between different laptop models.

2. Verify they are able to decrypt the persistent volume on the Journalist Workstation.
3. Ensure that they can connect to and login to the Journalist Interface.
4. Ensure that they have a Transfer Device, and access to its passphrase.
5. Verify they have access to the Secure Viewing Station by plugging in the Secure Viewing Station USB drive into the air-gapped computer designated for this purpose, booting, and verifying they can decrypt the persistent volume.
Note: It is especially important to only boot the Secure Viewing Station USB drive on the air-gapped computer designated for this purpose.

6. Verify the Submission Private Key is present in the Secure Viewing Station persistent volume by clicking the clipboard icon in the top right corner of the Tails desktop and selecting “Manage Keys”. When clicking “GnuPG keys” the key should be present.

Tip: The journalist should have all the credentials used in this checklist saved in the KeePassX database stored in the persistent volume of the Journalist Workstation.

7. If you are using a printer, verify that they are able to print a document from the Secure Viewing Station. If you are using an Export Device, verify that they are able to unlock the encrypted volume.

At this point, the journalist has verified they have the devices and credentials they need and can proceed to a walkthrough of the entire SecureDrop workflow.
SecureDrop is only as secure as the environment that surrounds it. To keep sources safe, the news organization’s website, physical space, and dedicated SecureDrop hardware must employ a set of basic security best practices or risk losing any source protection provided by SecureDrop.

Freedom of the Press Foundation eventually plans to list all of those SecureDrop onion URLs that meet the minimum requirements for deployment best practices as “verified” on its website. If your organization cannot follow the minimum guidelines, we cannot recommend your SecureDrop instance as safe to use.

In addition to implementing the following best practices, we strongly recommend that you have a reputable security firm perform a review of your organization’s public website prior to launching an instance of SecureDrop. Upon request, we can help put you in touch with a few security firms if you need more assistance.
SecureDrop itself runs as a Tor Onion Service. Organizations also need to create a SecureDrop Landing Page that will explain how SecureDrop works, give sources instructions on how to access the Tor Onion Service, and disclose the risks. We also recommend including a privacy policy (see our Sample SecureDrop Privacy Policy) describing what data is collected and how it will be used by your organization.

**Note:** SecureDrop will bring more attention to your organization from security researchers and others. A Landing Page that fails to implement minimum security requirements is sure to be noticed, and could undermine trust, discouraging possible sources.

### 23.1 URL and Location

Ideally you would not use a separate subdomain, but would use a path at your top-level domain, e.g. organization.com/securedrop. This is because TLS does not encrypt the hostname, so a SecureDrop user whose connection is being monitored would be trivially discovered.

If the Landing Page is deployed on the same domain as another site, you might consider having some specific configuration (such as the security headers below) apply only to the /securedrop request URI. This can be done in Apache by the encapsulating these settings within a `<Location>` block, which can be defined similarly in nginx by using the `location {}` directive.

### 23.2 HTTPS Only (No Mixed Content)

HTTPS encryption is the number-one security requirement for your site’s SecureDrop Landing Page. Without HTTPS, a source can easily be exposed as a visitor to your site.

This may be difficult if your website serves advertisements or utilizes a legacy content delivery network. You should make sure the SecureDrop Landing Page does not serve ads of any kind, even if the rest of your site does.
If you do not serve ads on any of your site, you should also consider switching your whole site over to HTTPS by default immediately. If you do serve ads, consider pressuring your ad networks to enable you to switch to HTTPS for your entire website in the future.

If your website needs to operate in both HTTPS and HTTP mode, use protocol-relative URLs for resources such as images, CSS and JavaScript in common templates to ensure your page does not end up in a mixed HTTPS/HTTP state. Consider submitting your domain to be included in the Chrome HSTS preload list if you can meet all of the requirements. This will tell web browsers that the site is only ever to be reached over HTTPS.

23.3 Perfect Forward Secrecy

Perfect Forward Secrecy (PFS) is a property of encryption protocols that ensures each SSL session has a unique key, meaning that if the key is compromised in the future it can’t be used to decrypt previously recorded SSL sessions. You may need to talk to your CA (certificate authority) and CDN (content delivery network) for this, although our recommended configuration below provides forward secrecy.

23.4 SSL Certificate Recommendations

Regardless of where you choose to purchase your SSL cert and which CA issues it, you’ll often be asked to generate the private key and a CSR (certificate signing request).

When you do this, it’s imperative that you use SHA-2 as the hashing algorithm instead of SHA-1, which is being phased out. You should also choose a key size of at least 2048 bits. These parameters will help ensure that the encryption used on your Landing Page is sufficiently strong. The following example OpenSSL command will create a private key and CSR with a 4096-bit key length and a SHA-256 signature:

```
openssl req -new -newkey rsa:4096 -nodes -sha256 -keyout domain.com.key -out domain.com.csr
```

Don’t load any resources (scripts, web fonts, etc.) from third parties (e.g. Google Web Fonts)

This will potentially leak information about sources to third parties, which can more easily be accessed by law enforcement agencies. Simply copy them to your server and serve them yourself to avoid this problem.

23.5 Do Not Use Third-Party Analytics, Tracking, or Advertising

Most news websites, even those that are non-profits, use third-party analytics tools or tracking bugs on their websites. It is vital that these are disabled for the SecureDrop Landing Page.

In the past, some news organizations were heavily criticized when launching their SecureDrop instances because their Landing Page contained trackers. They claimed they were going to great lengths to protect sources’ anonymity, but by having trackers on their Landing Page, this also opened up multiple avenues for third parties to collect information on those sources. This information can potentially be accessed by law enforcement or intelligence agencies and could unduly expose a source.

Similarly, consider avoiding Cloudflare (and other CDNs like Akamai, StackPath, Incapsula, Amazon CloudFront, etc.) for the SecureDrop Landing Page. These services intercept requests between a potential source and the SecureDrop Landing Page and can be used to track or collect information on sources.
23.6 Do Not Hyperlink .onion Addresses

Because a visitor to your Landing Page may not be using the Tor browser yet, clicking a link to your SecureDrop instance or to any other .onion address may result in an error message. Worse, depending on the browser and network configuration, it may cause lookups that an adversary can use to identify SecureDrop-related behavior.

Instead, we recommend including .onion addresses in plain text, without a hyperlink.

23.7 Avoid Direct Links to SecureDrop.org

We appreciate that you may want to link to the SecureDrop website to give Landing Page visitors more information about the system. Unfortunately, if a visitor visits these links without using the Tor browser, this generates traffic that an adversary may be able to use to identify SecureDrop-related behavior, regardless of the use of HTTPS.

We suggest offering a reference to the SecureDrop Onion Service in plain text, without a hyperlink (as per the preceding section): secrdrop5wyphb5x.onion

23.8 Apply Security Headers

Security headers give instructions to the web browser on how to handle requests from the web application. These headers set strict rules for the browser and help mitigate against potential attacks. Given the browser is a main avenue for attack, it is important these headers are as strict as possible.

You can use the site securityheaders.com to easily test your website’s security headers.

If you use Apache, you can use these:

<table>
<thead>
<tr>
<th>Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>set Cache-Control &quot;max-age=0, no-cache, no-store, must-revalidate&quot;</td>
</tr>
<tr>
<td>set Set-Cookie ^(.*)$ $;HttpOnly</td>
</tr>
<tr>
<td>set Pragma &quot;no-cache&quot;</td>
</tr>
<tr>
<td>set Expires &quot;-1&quot;</td>
</tr>
<tr>
<td>append X-Frame-Options: DENY</td>
</tr>
<tr>
<td>set X-XSS-Protection: &quot;1; mode=block&quot;</td>
</tr>
<tr>
<td>set X-Content-Type-Options: nosniff</td>
</tr>
<tr>
<td>set X-Download-Options: noopen</td>
</tr>
<tr>
<td>X-Permitted-Cross-Domain-Policies: master-only</td>
</tr>
<tr>
<td>Content-Security-Policy: &quot;default-src 'self'&quot;</td>
</tr>
<tr>
<td>Referrer-Policy &quot;no-referrer&quot;</td>
</tr>
</tbody>
</table>

If you intend to run nginx as your webserver instead, this will work:

<table>
<thead>
<tr>
<th>add_header</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache-Control &quot;max-age=0, no-cache, no-store, must-revalidate&quot;;</td>
</tr>
<tr>
<td>Pragma no-cache;</td>
</tr>
<tr>
<td>Expires -1;</td>
</tr>
<tr>
<td>X-Frame-Options DENY;</td>
</tr>
<tr>
<td>X-XSS-Protection &quot;1; mode=block&quot;;</td>
</tr>
<tr>
<td>X-Content-Type-Options nosniff;</td>
</tr>
<tr>
<td>X-Download-Options noopen;</td>
</tr>
<tr>
<td>X-Permitted-Cross-Domain-Policies master-only;</td>
</tr>
<tr>
<td>Content-Security-Policy &quot;default-src 'self'&quot;;</td>
</tr>
<tr>
<td>Referrer-Policy &quot;no-referrer&quot;;</td>
</tr>
</tbody>
</table>
23.9 Additional Apache Configuration

To enforce HTTPS/SSL always, you need to set up redirection within the HTTP (port 80) virtual host:

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
```

The same thing can be achieved in nginx with a single line:

```
return 301 https://$server_name$request_uri;
```

In your SSL (port 443) virtual host, set up HSTS and use these settings to give preference to the most secure cipher suites:

```
Header set Strict-Transport-Security "max-age=16070400;"
SSLProtocol all -SSLv2 -SSLv3
SSLHonorCipherOrder on
SSLCompression off
SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
```

Here’s a similar example for nginx:

```
add_header Strict-Transport-Security max-age=16070400;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_prefer_server_ciphers on;
ssl_ciphers "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH";
```

Here’s a similar example for nginx if the system supports TLS 1.3:

```
add_header Strict-Transport-Security max-age=16070400;
ssl_protocols TLSv1.2 TLSv1.3;
ssl_prefer_server_ciphers on;
```

Note: We have prioritized security in selecting these cipher suites, so if you choose to use them then your site might not be compatible with legacy or outdated browsers and operating systems. For a good reference check out Mozilla’s recommendations.

You’ll need to run `a2enmod headers ssl rewrite` for all these to work. You should also set `ServerSignature Off` and `ServerTokens Prod`, typically in `/etc/apache2/conf.d/security`. For nginx, use `server_tokens off;` so that the webserver doesn’t leak extra information.

If you use nginx, you can follow this link and use the configuration example provided by ProPublica.

23.9.1 Change detection monitoring for the web application configuration and Landing Page content

OSSEC is a free and open source host-based intrusion detection suite that includes a file integrity monitor. More information can be found here.
23.9.2 Don’t log access to the Landing Page in the webserver

Here’s an Apache example that would exclude the Landing Page from logging. However you still need to make sure no other assets get logged!

```bash
SetEnvIf Request_URI "^/securedrop($|/(\.*))" dontlog
CustomLog logs/access_log common env=!dontlog
```

In nginx, logging can be disabled by adding the following directives within the Landing Page location {} block:

```bash
access_log off;
error_log /dev/null;
```

23.10 Further Security Considerations

To guard your Landing Page against being modified by an attacker and directing sources to a rogue SecureDrop instance, you will need good security practices applying to the machine where it is hosted. Whether it’s a VPS in the cloud or dedicated server in your office, you should consider the following:

- Brute force login protection (see fail2ban or sshguard)
- Disable root SSH login
- Use SSH keys instead of passwords
- Use long, random and complex passwords
- Firewall rules to restrict accessible ports (see iptables or ufw)
- AppArmor, grsecurity, SELINUX, modsecurity
- Intrusion and/or integrity monitoring (see Logwatch, OSSEC, Snort, rkhunter, chkrootkit)
- Downtime alerts (Nagios or Pingdom)
- Two-factor authentication (see libpam-google-authenticator, libpam-yubico)

It’s preferable for the Landing Page to have its own segmented environment instead of hosting it alongside other sites running potentially vulnerable software or content management systems. Check that user and group file permissions are locked down and that modules or gateway interfaces for dynamic scripting languages are not enabled. You don’t want any unnecessary code or services running as this increases the attack surface.

23.11 Landing Page Content Suggestions

The content below presents sample text for the SecureDrop component of a news organization’s tips page. It does not account for any specific legal or organizational needs, but should provide guidance for any outlet getting started on crafting Landing Page language. Any tweaks to the sample content should be left to the legal and editorial discretion of the individual outlet, and should be viewed as essential to upholding source protection and transparency.

What is SecureDrop?

SecureDrop is an anonymity tool for journalists and whistleblowers. As a source, you can use our SecureDrop installation to anonymously submit documents to our organization. Our journalists use SecureDrop to receive source materials and securely communicate with anonymous contacts.

What should I know before submitting material through SecureDrop?
To protect your anonymity when using SecureDrop, it is essential that you do not use a network or device that can easily be traced back to your real identity. Instead, use public wifi networks and devices you control.

- Do NOT access SecureDrop on your employer’s network.
- Do NOT access SecureDrop using your employer’s hardware.
- Do NOT access SecureDrop on your home network.
- DO access SecureDrop on a network not associated with you, like the wifi at a library or cafe.

Got it. How can I submit files and messages through SecureDrop?

Once you are connected to a public network at a cafe or library, download and install the Tor Browser. Launch the Tor Browser. Visit our organization’s unique SecureDrop URL at http://our-unique-URL.onion/. Follow the instructions you find on our source page to send us materials and messages.

When you make your first submission, you will receive a unique codename. Memorize it. If you write it down, be sure to destroy the copy as soon as you’ve committed it to memory. Use your codename to sign back in to our source page, check for responses from our journalists, and upload additional materials.

As a source, what else should I know?

No tool can absolutely guarantee your security or anonymity. The best way to protect your privacy and anonymity as a source is to adhere to best practices.

You can use a separate computer you’ve designated specifically to handle the submission process. Or, you can use an alternate operating system like Tails, which boots from a USB stick and erases your activity at the end of every session.

A file contains valuable metadata about its source — when it was created and downloaded, what machine was involved, the machine’s owner, etc. You can scrub metadata from some files prior to submission using the Metadata Anonymization Toolkit featured in Tails.

Your online behavior can be extremely revealing. Regularly monitoring our publication’s social media or website can potentially flag you as a source. Take great care to think about what your online behavior might reveal, and consider using Tor Browser to mitigate such monitoring.

Our organization retains strict access control over our SecureDrop project. A select few journalists within our organization will have access to SecureDrop submissions. We control the servers that store your submissions, so no third party has direct access to the metadata or content of what you send us.

Do not discuss leaking or whistleblowing, even with trusted contacts.
Minimum requirements for the SecureDrop environment

- The Application and Monitor Servers should be dedicated physical machines, not virtual machines.
- A trusted location to host the servers. The servers should be hosted in a location that is owned or occupied by the organization to ensure that their legal department can not be bypassed with gag orders.
- The SecureDrop servers should be on a separate internet connection or completely segmented from corporate network.
- All traffic from the corporate network should be blocked at the SecureDrop’s point of demarcation.
- Video monitoring should be recorded of the server area and the organizations safe.
- Journalists should ensure that while using the air-gapped viewing station they are in an area without video cameras.
- An established monitoring plan and incident response plan. Who will receive the OSSEC alerts and what will their response plan be? These should cover technical outages and a compromised environment plan.
CHAPTER 25

Whole Site Changes

Ideally, some or all of the following changes are made to improve the overall security of the path to the Landing Page and obfuscate traffic analysis.

1. Make your entire site available through HTTPS.
   - That way, visits to your Landing Page won’t stand out as the only encrypted traffic to your site.
2. Include an iframe for all (or a random subset of) visitors, loading this particular URL (hidden).
   - By artificially generating traffic to the endpoint it will be harder to distinguish these from other, ‘real’ requests.
   - Use a random delay for adding the iframe (otherwise the ‘pairing’ with the initial HTTP request may distinguish this traffic).
3. Print the link, URL and info block on the dead trees (the paper), as others have suggested.
4. Add HSTS headers.

25.1 Suggested

- For publicly advertised SecureDrop instances display the Source Interface’s Onion Service onion address on all of the organization public pages.
- Mirror the Tor Browser and Tails so sources do not have to visit torproject.org to download it.
Sample SecureDrop Privacy Policy

[DATE]

SecureDrop strives to create a more secure environment for whistleblowers to give information to journalists. It was installed at [MEDIA ORG] with the help of Freedom of the Press Foundation.

Please read this privacy policy carefully. It explains what information SecureDrop does and does not collect, and why.

26.1 Collection of Information From Sources

- We don’t ask or require you to provide any personally identifying information when you submit materials through SecureDrop.

- The system does not record your IP address, information about your browser, computer, or operating system. Furthermore, the SecureDrop pages do not embed third-party content or deliver persistent cookies to your browser.

- The server will only store the date and time of the newest message sent from each source. Once you send a new message, the time and date of your previous message is automatically deleted.

- Journalists decrypt and read each message offline. They are encouraged to delete messages from the server on a regular basis. The date and time of any message will be securely deleted from the server when the message is deleted.

- Please keep in mind that the actual messages you send and receive through SecureDrop may include personally identifying information. For this reason, once you read a journalist’s message, we recommend you delete it. It will then be securely deleted from the file system.

Also please note that when you submit certain types of files through SecureDrop, you may be sending us metadata associated with that file.

For example, if you submit a photo through SecureDrop in JPEG format, the file may include information about the date, time, and the GPS location of where it was taken, and the type of device used to take the photo. Similarly, if you submit a Word file (.doc or .docx) through SecureDrop, it may include the identity of the document’s author,
the author’s operating system, GPS data about the author’s location, and the date and time when the document was created.

Our policy is to scrub metadata from the files we receive through SecureDrop before publication. If you don’t want to send us metadata, please use the Metadata Anonymization Toolkit to scrub the file before you submit it.

26.2 Collection of Information About Journalists’ Use of SecureDrop

[MEDIA ORG] collects information about journalists’ use of SecureDrop for security monitoring and to make sure the system works properly.

This information we collect about journalists includes details about the device, browser, and operating system journalists use when accessing the system, and the date and time of each session.

We retain these access logs for [____] days, and then delete them.

26.3 Data Security

[MEDIA ORG] works diligently to protect the identities of our sources and keep the information they give us confidential.

SecureDrop servers are under the physical control of [MEDIA ORG] and do not share common elements of the [MEDIA ORG’S] other infrastructure.

However, no one can truly guarantee 100% security of any system. Like all software, SecureDrop may contain bugs. Ultimately, you use the SecureDrop service at your own risk.

26.4 Children Under 13

The Children’s Online Privacy Protection Act restricts our ability to collect personal information from children under 13. This site is not directed to children 12 or younger.

26.5 Changes to This Policy

We may revise this Privacy Policy from time to time. The most current version of the policy will govern our collection and use of personal information and will always be at [LINK]. If we make changes that we believe are material, we will prominently display a notice on our site [____] days before we make those changes.

26.6 Contact

[MEDIA ORG] welcomes questions, concerns, and feedback about this policy. If you have suggestions for us, feel free to let us know at [EMAIL ADDRESS].
This is the minimum hardware that must be acquired to install SecureDrop:

- 2 computers with memory and hard drives to use as the SecureDrop servers.
- Mouse, keyboard, monitor (and necessary dongle or adapter) for installing the servers.
- At least 2 dedicated physical computers that can boot to Tails: one computer for the Secure Viewing Station, and one or more computers for the Admin Workstation(s)/Journalist Workstation(s).
- Dedicated airgapped hardware for the mouse, keyboard, and monitor (only if you are using a desktop for the Secure Viewing Station).
- Network firewall.
- At least 3 ethernet cables.
- Plenty of USB sticks: 1 drive for the master Tails stick, 1 drive for each Secure Viewing Station, 1 drive for each Transfer Device, 1 drive for each Export Device, and 1 drive for each admin and journalist.

Additionally, you may want to consider the following purchases:

- a printer without wireless network support, to use in combination with the Secure Viewing Station.
- an external hard drive to expand the storage capacity of the Secure Viewing Station.
- an external hard drive for server backups.
- a USB drive to store backups of your Tails workstation drives.
- a network switch, if you use a firewall with fewer than four ports.
- a hardware token for HOTP authentication, such as a YubiKey, if you want to use hardware-based two-factor authentication instead of a mobile app.
- a USB drive with a physical write protection switch, or a USB write blocker, if you want to mitigate the risk of introducing malware from your network to your Secure Viewing Station during repeated use of an Export Device.
- CD-R/DVD-R writers, if you want to use CD-Rs/DVD-Rs as transfer or export media, and a CD shredder that can destroy media consistent with your threat model.
Important: To avoid hardware compatibility issues, we strongly recommend adhering to our specific hardware recommendations. Please review the Hardware section in full for further details.
Promoting Your SecureDrop Instance

At Freedom of the Press Foundation, we’ve found news organizations that get the most out of SecureDrop are those who promote it regularly and effectively. SecureDrop will only be used by sources if they know it exists, so it’s best to promote its use in a variety of ways so that a wide swath of people will see it.

So here are a few tips used by some of the news outlets that have seen the most success with SecureDrop.

28.1 Make a High Profile Announcement

Anytime you launch a SecureDrop, you’ll want to write an accompanying news story along with it to alert your readers and potential sources where to submit information. Almost every news organization already does this, but some good recent examples come from USA Today, The Guardian, and Wired. You can also write a companion Q & A like the Washington Post did.

However, a launch announcement is really just a small piece of the puzzle. It’s important to regularly remind readers and potential sources that your SecureDrop exists, because only a tiny fraction will likely see the launch announcement and it will quickly be buried in other news after a couple of days.

28.2 Provide a Clear Link on Your Homepage

Making your SecureDrop or secure tips page easy to find is one of the most important things you can do to ensure that potential sources use it. The best way you can do this is providing a clear link on your homepage, so that every time a user goes to your website, they can quickly see where they need to go.

For example, the Intercept has a “become a source” link in its main menu:
The Washington Post has a link on their front page for “how to share a tip securely”: 
Other news organizations put a little link in their footer, however, we’ve found that this is not as effective as putting it in a more prominent on your front page.

28.3 Provide Links at the Bottom of Your Articles

Another great way to remind potential sources know that they can use SecureDrop is to put a link at the bottom of each article. For example, Gizmodo Media Group, uses a message like this:

Have something you think we should know? Email us at tips@deadspin.com, call our confidential tips hotline at (347) 746-8471, or contact our writers directly, or use our SecureDrop system. You can also follow us on Twitter, like us on Facebook, and sign up for our newsletter!

28.4 Create an Instructional Video on How to Access and Use Your SecureDrop

To better help potential sources visualize how SecureDrop works, several organizations have made short instructional videos walking through all the steps. Some good examples include the Toronto Globe and Mail, The Intercept, and Lucy Parsons Labs.

28.5 Regularly Share Your SecureDrop Landing Page on Social Media

The majority of adults in the United States now get their news from Facebook or other social media sites like Twitter, so it’s important to regularly remind people via social media posts that SecureDrop is the safest way they can contact your journalists if they have a sensitive tip to share. If there’s specific stories you are looking for tips on that may already be in the news, this is a great way of getting added attention to your SecureDrop.
28.6 Target Potential Whistleblowers with Advertising

Facebook and Twitter also allow for targeted advertising to users in specific locations, attributes, and sometimes even specific users. Gizmodo Media Group recently targeted online advertisements for their secure tips page at DC residents imploring them to tell on trump. At Freedom of the Press Foundation, we ran a proof of concept Twitter advertisement aimed at EPA and NOAA employees to show how it can be done. You can read about how you can do the same thing here.

28.7 Put an Advertisement in Your Physical Paper

Obviously this tip only applies to news outlets that also print a physical newspaper, but putting an ad or in the paper to tell readers where to go to access SecureDrop can be extremely effective.

The New York Times took out a full page ad in their own paper when they launched SecureDrop and other secure communications tools for their tips line:
So excited and proud to see @nytimes run a full page ad letting readers know how to securely send tips.
And the Toronto Globe and Mail regularly puts a note in their physical paper reminding potential sources where they can go:

Canadian news organization @globeandmail includes the URL to its @SecureDrop landing page in the printed paper:
SecureDrop attempts to solve or mitigate several problems journalists and sources have faced in recent legal investigations, attacks from state actors, and other threats to the confidentiality of communications.

29.1 No Third Parties that Can Secretly be Subpoenaed

For decades, there were very few leak prosecutions in the United States in large part because the government would have to subpoena reporters to testify against a source to get a conviction. That proved incredibly difficult, if not impossible, when reporters regularly refused to testify and threatened to go to jail rather than betray a source.

More recently, there have been a record number of leak prosecutions largely because the government has learned they don’t need reporters to testify against their sources anymore. Instead, they can just secretly subpoena third-party services like Google or AT&T or Verizon or Facebook and get a treasure trove of digital information on reporters and sources’ communications. For example, the Associated Press had twenty of their phone lines subpoenaed without their knowledge in order to identify a source. The government also got a warrant for Fox News reporter James Rosen’s Gmail account without him knowing. In both cases, their alleged sources were prosecuted, even though journalists never directly divulged their sources.

SecureDrop completely eliminates third parties from the equation and puts the power to challenge such cases back in the hands of reporters. The journalist and source communicate exclusively through one server that the news organization owns and sits on their property, so any legal order for information must go directly to the news organization rather than Google or AT&T. The news organization again has the power to contest the order or refuse to comply if they so wish.

29.2 Limits the Metadata Trail as Much as Possible

In many leak cases, the metadata of a journalist’s communications—where you’re located, who you’re talking to, when you’re talking to them, and how often—can lead to trouble just as much as the actual content of your conversations.

Even if a government serves a court order directly to a news organization to compel the disclosure of information, SecureDrop logs much less information than email providers or phone companies do.
The source can only log into SecureDrop through the Tor Browser, which masks the source’s IP address to begin with, so there is no indication who the source is (unless they disclose it) and where they are sending information from. The Tor IP address, the computer, and the browser type that the source is using is not logged either.

For each source, only the time and date of each submission is logged on the server. When a source sends a new message, the time and date of the last message is overwritten. This means that there won’t be a trail of metadata showing exactly when the source and journalist were talking.

In addition, sources cannot create a custom username that could reveal information about them. Instead, SecureDrop automatically generates two random codenames, one to show to the source and another to the journalists using the system.

### 29.3 Encrypted and Air-Gapped

Communications through SecureDrop are both encrypted in transit, so messages cannot be easily intercepted and read while they are traversing the Internet and are also encrypted on the server so if any attacker manages to break into the server, they would not be able to read past messages.

In addition, the decryption key for SecureDrop submissions sits on an air-gapped computer (not connected to the Internet). This air-gapped computer is the only place SecureDrop submissions are decrypted and read so that they are much harder for an attacker to access.

### 29.4 Protects Against Hackers

A 2014 study showed that 21 of the top 25 news organization had, at one time or another, been targeted by state sponsored hackers.

Because of this threat, SecureDrop completely segments its traffic from a news organization’s normal network. Submissions are accessed and downloaded using the Tails operating system, which boots off of a USB, does not touch your computer’s hard drive, and routes all its Internet traffic through Tor.

Submissions are decrypted on an air-gapped computer also using Tails. This mitigates against the risk that an attacker could send malware through SecureDrop in an attempt to infect the news organization’s normal network as well.

The SecureDrop servers also undergo significant system hardening in order to make it as difficult as possible for hackers to break in. By doing so, SecureDrop protects sources against networks that are already compromised, as well as a news organization’s normal network from attacks that could potentially come through SecureDrop.

### 29.5 Free and Open Source Software

100% of SecureDrop’s code is free and open source. Not only does this mean anyone can install SecureDrop themselves, but the code is available online for security experts to test for vulnerabilities.

SecureDrop has gone through four audits by third-party penetration testing firms and will continue to go through audits when major changes are made to the code base in the future. We always publish these audits publicly so everyone can be assured that SecureDrop is as safe to use as possible.
Investigating Logs

When troubleshooting issues with your SecureDrop instance, be sure to examine all relevant log files on both servers. To work with logs, it is helpful to be familiar with commands like `less`, `tail`, and `grep`; to inspect older, archived logs (names end with `.gz`) you can use commands like `zless` and `zgrep` to avoid manually decompressing each file.

**Note:** You can use the `securedrop-admin` tool to extract logs to send to Freedom of the Press Foundation for analysis. Run the following command on your Admin Workstation:

```
cd ~/Persistent/securedrop
./securedrop-admin logs
```

This command will produce encrypted tarballs containing logs from each server. See the command output for more information.

### 30.1 Logs to examine on both servers

- `/var/log/kern.log`: Use this file to investigate kernel-related issues, including warnings or errors specific to AppArmor or grsecurity (a set of patches applied to the kernels for additional security hardening)

- `/var/log/syslog`: Use this file to investigate most other system issues, including iptables configuration problems or Tor network issues. Use search patterns, e.g., search for “app Tor” to find log entries specific to Tor.

### 30.2 Application Server Logs

See the directory `/var/log/apache2/*` for web server access and error logs. In production systems, logging is only enabled for the *Journalist Interface* to the files `journalist-access.log` and `journalist-error.log`, and the logs do not contain IP address information.
When investigating an application error on the Source Interface (e.g., if you see an “Internal Server Error” when submitting a document), it can make sense to temporarily enable error logging. To do so:

1. Log into your Application Server from your Admin Workstation via ssh
2. Edit the file /etc/apache2/sites-enabled/source.conf (requires sudo)
3. Comment out the old ErrorLog and LogLevel directives, e.g., like so:

```
# Enabling logging for error investigation, 2020-04-18, ~admin
# ErrorLog /dev/null
# LogLevel critical
```

4. Add the desired new logging configuration in the same location (inside the `<VirtualHost>` block), e.g.:

```
ErrorLog /var/log/apache2/source-error.log
LogLevel debug
```

5. Save the file and reload the configuration with sudo service apache2 reload
6. Visit the Source Interface and reproduce the error
7. Inspect the log file /var/log/apache2/source-error.log for any details
8. Remember to set the configuration back to the default values once your investigation is complete.

Note that the debug logging level is highly verbose; if you want to adjust it, see the Apache documentation for more information about the different logging levels.

If you encounter an application error, and you have not modified the application code, please be sure to file an issue or contact us via securedrop@freedom.press (GPG encrypted).

### 30.3 Monitor Server Logs

- `/var/ossec/logs/ossec.log`: Examine this file to investigate problems with OSSEC itself not functioning as expected (e.g., you are not seeing alerts when you would expect them to).
- `/var/ossec/logs/alerts/alerts.log`: This file contains the most recent alerts generated by OSSEC. If you have correctly configured OSSEC emails, the text of these alerts should correspond to the text of the emails.
- `/var/log/mail.log` and `/var/log/procmail.log`: These files contain information about email delivery and email processing (for encrypting the alerts). Investigate these files if you believe OSSEC is correctly configured, but you are not receiving emails.
31.1 Setting Up OSSEC Alerts

OSSEC is an open source host-based intrusion detection system (IDS) that we use to perform log analysis, file integrity checking, policy monitoring, rootkit detection and real-time alerting. It is installed on the Monitor Server and constitutes that machine’s main function. OSSEC works in a server-agent scheme, that is, the OSSEC server extends its existing functions to the Application Server through an agent installed on that server, covering monitoring for both machines.

In order to receive email alerts from OSSEC, you need to supply several settings to Ansible in the playbook for your environment. If you don’t already have a working mail server or don’t know what to do, then see the section below about using Gmail as a fallback option. We assume that you’re working out of the ‘securedrop’ directory you cloned the code into, and editing `install_files/ansible-base/group_vars/all/site-specific` prior to installing SecureDrop.

What you need:

- The GPG key that OSSEC will encrypt alerts to
- The email address that will receive alerts from OSSEC
- Information for your SMTP server or relay (hostname, port)
- Credentials for the email address that OSSEC will send alerts from

Receiving email alerts from OSSEC requires that you have an SMTP relay to route the emails. You can use an SMTP relay hosted internally, if one is available to you, or you can use a third-party SMTP relay such as Gmail. The SMTP relay does not have to be on the same domain as the destination email address, i.e. smtp.gmail.com can be the SMTP relay and the destination address can be `securedrop@freedom.press`.

While there are risks involved with receiving these alerts, such as information leakage through metadata, we feel the benefit of knowing how the SecureDrop servers are functioning is worth it. If a third-party SMTP relay is used, that relay will be able to learn information such as the IP address the alerts were sent from, the subject of the alerts, and the destination email address the alerts were sent to. Only the body of an alert email is encrypted with the recipient’s GPG key. A third-party SMTP relay could also prevent you from receiving any or specific alerts.
The SMTP relay that you use should support SASL authentication and SMTP TLS protocols TLSv1.2, TLSv1.1, and TLSv1. Most enterprise email solutions should be able to meet those requirements.

Below are the values you must specify in to configure OSSEC correctly. For first-time installs, you can use the configuration playbook, or edit install_files/ansible-base/group_vars/all/site-specific manually.

- GPG public key used to encrypt OSSEC alerts: ossec_alert_gpg_public_key
- Fingerprint of key used when encrypting OSSEC alerts: ossec_gpg_fpr
- The email address that will receive alerts from OSSEC: ossec_alert_email
- The reachable hostname of your SMTP relay: smtp_relay
- The secure SMTP port of your SMTP relay: smtp_relay_port (typically 25, 587, or 465. must support TLS encryption)
- Email username to authenticate to the SMTP relay: sasl_username
- Domain name of the email used to send OSSEC alerts: sasl_domain
- Password of the email used to send OSSEC alerts: sasl_password

If you don’t know what value to enter for one of these, please ask your organization’s email admin for the full configuration before proceeding. It is better to get these right the first time rather than changing them after SecureDrop is installed. If you’re not sure of the correct smtp_relay_port number, you can use a simple mail client such as Thunderbird to test different settings or a port scanning tool such as nmap to see what’s open. You could also use telnet to make sure you can connect to an SMTP server, which will always transmit a reply code of 220 meaning “Service ready” upon a successful connection.

The smtp_relay mail server hostname is often, but not always, different from the sasl_domain, e.g. smtp.gmail.com and gmail.com.

In some cases, authentication or transport encryption mechanisms will vary and you may require later edits to the Postfix configuration (mainly /etc/postfix/main.cf) on the Monitor Server in order to get alerts to work. You can consult Postfix’s official documentation for help, although we’ve described some common scenarios in the troubleshooting section.

If you have your GPG public key handy, copy it to install_files/ansible-base and then specify the filename, e.g. ossec.pub, in the ossec_alert_gpg_public_key line of group_vars/all/site-specific.

If you don’t have your GPG key ready, you can run GnuPG on the command line in order to find, import, and export your public key. It’s best to copy the key from a trusted and verified source, but you can also request it from keyservers using the known fingerprint. Looking it up by email address or a shorter key ID format could cause you to obtain a wrong, malicious, or expired key. Instead, we recommend you type out your fingerprint in groups of four (just like GPG prints it) enclosed by double quotes. The reason we suggest this formatting for the fingerprint is simply because it’s easiest to type and verify correctly. In the code below simply replace <fingerprint> with your full, space-separated fingerprint:

```
gpg --recv-key "<fingerprint>"
```

**Note:** It is important you type this out correctly. If you are not copy-pasting this command, we recommend you double-check you have entered it correctly before pressing enter.

Again, when passing the full public key fingerprint to the --recv-key command, GPG will implicitly verify that the fingerprint of the key received matches the argument passed.
Caution: If GPG warns you that the fingerprint of the key received does not match the one requested do not proceed with the installation. If this happens, please email us at securedrop@freedom.press.

Next we export the key to a local file.

```
gpg --export -a "<fingerprint>" > ossec.pub
```

Copy the key to a directory where it’s accessible by the SecureDrop installation:

```
cp ossec.pub install_files/ansible-base/
```

The fingerprint is a unique identifier for an encryption (public) key. The short and long key ids correspond to the last 8 and 16 hexadecimal digits of the fingerprint, respectively, and are thus a subset of the fingerprint. The value for ossec_gpg_fpr must be the full 40 hexadecimal digit GPG fingerprint for this same key, with all capital letters and no spaces. The following command will retrieve and format the fingerprint per our requirements:

```
gpg --with-colons --fingerprint "<fingerprint>" | grep "^fpr" | cut -d: -f10
```

Next you specify the e-mail that you’ll be sending alerts to, as ossec_alert_email. This could be your work email, or an alias for a group of IT admins at your organization. It helps for your mail client to have the ability to filter the numerous messages from OSSEC into a separate folder.

Now you can move on to the SMTP and SASL settings, which are straightforward. These correspond to the outgoing e-mail address used to send the alerts instead of where you’re receiving them. If that e-mail is ossec@news-org.com, the sasl_username would be ossec and sasl_domain would be news-org.com.

The Postfix configuration enforces certificate verification, and requires both a valid certificate and STARTTLS support on the SMTP relay. By default the system CAs will be used for validating the relay certificate. If you need to provide a custom CA to perform the validation, copy the cert file to install_files/ansible-base add a new variable to group_vars/all/site-specific:

```
smtp_relay_cert_override_file: MyOrg.crt
```

where MyOrg.crt is the filename. The file will be copied to the server in /etc/ssl/certs_local and the system CAs will be ignored when validating the SMTP relay TLS certificate.

Save group_vars/all/site-specific, exit the editor and proceed with the installation by running the playbooks.

### 31.1.1 Using Gmail for OSSEC Alerts

It’s easy to get SecureDrop to use Google’s servers to deliver the alerts, but it’s not ideal from a security perspective. This option should be regarded as a backup plan. Keep in mind that you’re leaking metadata about the timing of alerts to a third party — the alerts are encrypted and only readable to you, however that timing may prove useful to an attacker.

First you should sign up for a new account. While it’s technically possible to use an existing Gmail account, it’s best to compartmentalize these alerts from any of your other activities. Choose a strong and random passphrase for the new account. Skip the creation of a Google+ profile and continue straight to Gmail. Next, enable Google’s 2-Step Verification. With 2-Step Verification enabled, you won’t use the normal account password in this configuration — it will not work; instead you must navigate (using the settings in the top right) to Account > Signing in > App passwords, and generate a new App password which you will use as the sasl_passwd.

Once the account is created you can log out and provide the values for sasl_username as your new Gmail username (without the domain), sasl_domain, which is typically gmail.com (or your custom Google Apps domain), and...
sasl_passwd. Remember to use the App password generated from the 2-step config for sasl_passwd, as the primary account password won’t work. The smtp_relay is smtp.gmail.com and the smtp_relay_port is 587.

### 31.1.2 Configuring Fingerprint Verification

If you run your own mail server, you may wish to increase the security level used by Postfix for sending mail to fingerprint, rather than secure. Doing so will require an exact match for the fingerprint of TLS certificate on the SMTP relay. The advantage to fingerprint verification is additional security, but the disadvantage is potential maintenance cost if the fingerprint changes often. If you manage the mail server and handle the certificate rotation, you should update the SecureDrop configuration whenever the certificate changes, so that OSSEC alerts continue to send.

Using fingerprint verification does not work well for popular mail relays such as smtp.gmail.com, as those fingerprints can change frequently, due to load balancing or other factors.

You can retrieve the fingerprint of your SMTP relay by running the command below (all on one line). Please note that you will need to replace smtp.gmail.com and 587 with the correct domain and port for your SMTP relay.

```
openssl s_client -connect smtp.gmail.com:587 -starttls smtp < /dev/null 2>/dev/null | openssl x509 -fingerprint -noout -in /dev/stdin | cut -d'=' -f2
```

If you are using Tails, you will not be able to connect directly with `openssl s_client` due to the default firewall rules. To get around this, proxy the requests over Tor by adding `torify` at the beginning of the command. The output of the command above should look like the following:

```
```

Finally, add a new variable to `group_vars/all/site-specific` as smtp_relay_fingerprint, like so:

```
```

Specifying the fingerprint will configure Postfix to use it for verification on the next playbook run. (To disable fingerprint verification, simply delete the variable line you added, and rerun the playbooks.) Save `group_vars/all/site-specific`, exit the editor and proceed with the installation by running the playbooks.

### 31.2 Troubleshooting

Some OSSEC alerts should begin to arrive as soon as the installation has finished.

The easiest way to test that OSSEC is working is to SSH to the Monitor Server and run `service ossec restart`. This will trigger an Alert level 3 saying: “Ossec server started.”

So you’ve finished installing SecureDrop, but you haven’t received any OSSEC alerts. First, check your spam/junk folder. If they're not in there, then most likely there is a problem with the email configuration. In order to find out what’s wrong, you’ll have to SSH to the Monitor Server and take a look at the logs. To examine the mail log created by Postfix, run the following command:

```
tail /var/log/mail.log
```

The output will show you attempts to send the alerts and provide hints as to what went wrong. Here’s a few possibilities and how to fix them:
<table>
<thead>
<tr>
<th>Problem</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection timed out</td>
<td>Check that the hostname and port is correct in the relayhost line of /etc/postfix/main.cf</td>
</tr>
<tr>
<td>Server certificate not verified</td>
<td>Check that the relay certificate is valid (for more detailed help, see Troubleshooting SMTP TLS). Consider adding smtp_relay_cert_override_file to prod_specific.yml as described above.</td>
</tr>
<tr>
<td>Authentication failure</td>
<td>Edit /etc/postfix/sasl_passwd and make sure the username, domain and password are correct. Run postmap /etc/postfix/sasl_passwd to update when finished.</td>
</tr>
</tbody>
</table>

After making changes to the Postfix configuration, you should run service postfix reload and test the new settings by restarting the OSSEC service.

**Tip:** If you change the SMTP relay port after installation for any reason, you must update the smtp_relay_port variable in the group_vars/all/site-specific file, then rerun the Ansible playbook. As a general best practice, we recommend modifying and rerunning the Ansible playbook instead of manually editing the files live on the servers, since values like smtp_relay_port are used in several locations throughout the config.

### 31.2.1 Useful Log Files for OSSEC

Other log files that may contain useful information:

- **/var/log/procmail.log**  Includes lines for sending mail containing OSSEC alerts.
- **/var/log/syslog**  Messages related to grsecurity, AppArmor and iptables.
- **/var/ossec/logs/ossec.log**  OSSEC’s general operation is covered here.
- **/var/ossec/logs/alerts/alerts.log**  Contains details of every recent OSSEC alert.

**Tip:** Remember to encrypt any log files before sending via email, for example to securedrop@freedom.press, in order to protect security-related information about your organization’s SecureDrop instance.

### 31.2.2 Not Receiving Emails

Some mail servers require that the sending email address match the account that authenticated to send mail. By default the Monitor Server will use ossec@ossec.server for the from line, but your mail provider may not support the mismatch between the domain of that value and your real mail host. If the Admin email address (configured as
ossec_alert_email in group_vars/all/site-specific) does not start receiving OSSEC alerts updates shortly after the first playbook run, try setting ossec_from_address in group_vars/all/site-specific to the full email address used for sending the alerts, then run the playbook again.

31.2.3 Message Failed to Encrypt

If OSSEC cannot encrypt the alert to the GPG public key for the Admin email address (configured as ossec_alert_email in group_vars/all/site-specific), the system will send a static message instead of the scheduled alert:

Failed to encrypt OSSEC alert. Investigate the mailing configuration on the Monitor Server.

Check the GPG configuration vars in group_vars/all/site-specific. In particular, make sure the GPG fingerprint matches that of the public key file you exported.

31.2.4 Troubleshooting SMTP TLS

Your choice of SMTP relay server must support STARTTLS and have a valid server certificate. By default, the Monitor Server’s Postfix configuration will try to validate the server certificate using the default root store (in Ubuntu, this is maintained in the ca-certificates package). You can override this by setting smtp_relay_cert_override_file as described earlier in this document.

In either situation, it can be helpful to use the openssl command line tool to verify that you can successfully connect to your chosen SMTP relay securely. We recommend doing this before running the playbook, but it can also be useful as part of troubleshooting OSSEC email send failures.

In either case, start by attempting to make a STARTTLS connection to your chosen smtp_relay:smtp_relay_port (get the values from your group_vars/all/site-specific file). On a machine running Ubuntu, run the following openssl command, replacing smtp_relay and smtp_relay_port with your specific values:

```bash
openssl s_client -showcerts -starttls smtp -connect smtp_relay:smtp_relay_port < /dev/null 2> /dev/null
```

Note that you will not be able to run this command on the Application Server because of the firewall rules. You can run it on the Monitor Server, but you will need to run it as the Postfix user (again, due to the firewall rules):

```bash
sudo -u postfix openssl s_client -showcerts -starttls smtp -connect smtp.gmail.com:587 < /dev/null 2> /dev/null
```

If the command fails with “Could not connect” or a similar message, then this mail server does not support STARTTLS. Verify that the values you are using for smtp_relay and smtp_relay_port are correct. If they are, you should contact the admin of that relay and talk to them about supporting STARTTLS, or consider using another relay that already has support.

If the command succeeds, the first line of the output should be “CONNECTED” followed by a lot of diagnostic information about the connection. You should look for the line that starts with “Verify return code”, which is usually one of the last lines of the output. Since we did not give openssl any information about how to verify certificates in the previous command, it should be a non-zero value (indicating verification failed). In my case, it is Verify return code: 20 (unable to get local issuer certificate), which indicates that openssl does not know how to build the certificate chain to a trusted root.

If you are using the default verification setup, you can check whether your cert is verifiable by the default root store with -CApath:
openssl s_client -CApath /etc/ssl/certs -showcerts -starttls smtp -connect smtp_relay:smtp_relay_port < /dev/null 2> /dev/null

For example, if I’m testing Gmail as my SMTP relay (smtp.gmail.com:587), running the openssl with the default root store results in Verify return code: 0 (ok) because their certificate is valid and signed by one of the roots in the default store. This indicates that can be successfully used to securely relay email in the default configuration of the Monitor Server.

If your SMTP relay server does not successfully verify, you should use the return code and its text description to help you diagnose the cause. Your cert may be expired, in which case you should renew it. It may not be signed by a trusted CA, in which case you should obtain a signature from a trusted CA and install it on the mail server. It may not have the right hostnames in the Common Name or Subject Alternative Names, in which case you will need to generate a new CSR with the correct hostnames and then obtain a new certificate and install it. Etc., etc.

If you are not using the the default verification setup, and intentionally do not want to use a certificate signed by one of the default CA’s in Ubuntu, you can still use openssl to test whether you can successfully negotiate a secure connection. Begin by copying your certificate file (smtp_relay_cert_override_file from group_vars/all/site-specific) to the computer you are using for testing. You can use -CAfile to test if your connection will succeed using your custom root certificate:

openssl s_client -CAfile /path/to/smtp_relay_cert_override_file -showcerts -starttls smtp_relay:smtp_relay_port < /dev/null 2> /dev/null

Finally, if you have a specific server in mind but are not sure what certificate you need to verify the connection, you can use the output of openssl s_client to figure it out. Since we have -showcerts turned on, openssl prints the entire certificate chain it receives from the server. A properly configured server will provide all of the certificates in the chain up to the root cert, which needs to be identified as “trusted” for the verification to succeed. To see the chain, find the part of the output that start with Certificate chain. It will look something like this (example from smtp.gmail.com, with certificate contents snipped for brevity):

---
Certificate chain
0 s:/C=US/ST=California/L=Mountain View/O=Google Inc/CN=smtp.gmail.com
i:/C=US/O=Google Inc/CN=Google Internet Authority G2
   -----BEGIN CERTIFICATE-----
   <snip>
   -----END CERTIFICATE-----
1 s:/C=US/O=Google Inc/CN=Google Internet Authority G2
i:/C=US/O=GeoTrust Inc./CN=GeoTrust Global CA
   -----BEGIN CERTIFICATE-----
   <snip>
   -----END CERTIFICATE-----
2 s:/C=US/O=GeoTrust Inc./CN=GeoTrust Global CA
i:/C=US/O=Equifax/OU=Equifax Secure Certificate Authority
   -----BEGIN CERTIFICATE-----
   <snip>
   -----END CERTIFICATE-----
---

The certificates are in reverse order from leaf to root. openssl handily prints the Subject (s:) and Issuer (i:) information for each cert. In order to find the root certificate, look at the Issuer of the last certificate. In this case, that’s Equifax Secure Certificate Authority. This is the root certificate that issued the first certificate in the chain, and it is what you need to tell Postfix to use in order to trust the whole connection.

Actually obtaining this certificate and establishing trust in it is beyond the scope of this document. Typically, if you are using your own SMTP relay with a custom CA, you will be able to obtain this certificate from an intranet portal or someone on your IT staff. For a well-known global CA, you can obtain it from the CA’s website. For example,
a quick search for “Equifax Secure Certificate Authority” finds the web page of GeoTrust’s Root Certificates, which have accompanying background information and are available for download.

Once you have the root certificate file, you can use `-CAfile` to test that it will successfully verify the connection.

### 31.3 Analyzing the Alerts

Understanding the contents of the OSSEC alerts requires a background and knowledge in Linux systems administration. They may be confusing, and at first it will be hard to tell between a genuine problem and a fluke. You should examine these alerts regularly to ensure that the SecureDrop environment has not been compromised in any way, and follow up on any particularly concerning messages with direct investigation.

#### 31.3.1 Common OSSEC Alerts

The SecureDrop Application and Monitor Servers reboot every night, as part of the unattended upgrades process. Therefore, on nights where packages were updated, you should receive email alerts every morning indicating binaries have changed. Below is a sample alert, but you may see any number of these records in the logs. This will happen in batches so these emails might be longer than the below alert. You should also see them in an email named `Daily Report: File Changes`. To verify this activity matches the package history, you can review the logs in `/var/log/apt/history.log`.

```
Received From: (app)
Rule: 2902 fired (level 7) -> "New (Debian Package) installed."
Portion of the log(s):
status installed <package name> <version>
```

These are normal alerts, they tell you your system is up-to-date and patched.

Occasionally your SecureDrop Servers will send an alert for failing to connect to Tor relays. Since SecureDrop runs as a Tor Onion Service, it is possible for Tor connections to timeout or become overloaded.

```
Received From: (app)
Rule: 1002 fired (level 2) -> "Unknown problem somewhere in the system."
Portion of the log(s):
[warn] Your Guard <name> ($fingerprint) is failing a very large amount of circuits. Most likely this means the Tor network is overloaded, but it could also mean an attack against you or potentially the guard itself.
```

This alert is common but if you see them for sustained periods of time (several times a day), please contact us at the SecureDrop Support Portal or at securedrop@freedom.press for help.

#### 31.3.2 Uncommon OSSEC Alerts

SecureDrop also runs automatic checks for submission data integrity problems. For example, secure deletion of large submissions could potentially be interrupted, resulting in an alert recommending steps to clean them up.

If you believe that the system is behaving abnormally, you should contact us at the SecureDrop Support Portal or securedrop@freedom.press for help.
Because Tails is supposed to be as amnesiac as possible, you want to shield your Tails stick from any extra inputs from, and outputs to, a potentially untrusted network. This is why we strongly recommend using a printer that does not have WiFi or Bluetooth, and hooking up to it using a regular USB cable to print.

Finding a printer that works with Tails can be challenging because Tails is based on the Linux operating system, which often has second-class hardware support in comparison to operating systems such as Windows or macOS.

We maintain a list of printers that we have personally tested and gotten to work with Tails, in the Hardware guide; if possible, we recommend using one of those printers. The Linux Foundation also maintains the OpenPrinting database, which documents the compatibility, or lack thereof, of numerous printers from almost every manufacturer.

Note: The latest generations of printers might or might not be represented by the OpenPrinting database; also, the database does not document whether or not a printer is wireless, so this will involve manually checking models of interest, if you wish to use this resource as a guide for purchasing a non-wireless printer suitable for use with SecureDrop.

With that in mind, this database is arguably the best resource for researching the compatibility of printers with Linux. As a tip for narrowing down your search, look for printers that are compatible with Debian, or Debian-based distributions like Ubuntu, since Tails itself is also Debian-based. This might increase the chances for a seamless installation experience in Tails.

In any case, this document outlines the usual set of steps that we follow when attempting to use a new printer with Tails.

Note: While, as of Tails 3, it’s no longer necessary to have admin privileges in order to install or configure printers, we recommend that you set an admin passphrase and unlock your persistent volume; this ensures that the printer’s installation and configuration settings persist after every reboot, so you don’t have to reinstall it each time you start Tails.
32.1 Installing and Printing via the Tails GUI

Let’s look at Tails 3.0’s typical flow for installing a USB-connected printer. If you’ve enabled persistence, boot with your persistent volume, and set an admin passphrase. Connect the printer to your Tails-booted computer via USB, then turn the printer on.

Now, you’ll want to single-click your way through Applications System Tools Settings Printers.

In this example, we’ll assume that this is the first time we’ve tried to install a printer, which will show the following:
Click **Add a Printer**. By doing so, you’ll now get a list of printers that Tails has auto-detected. You should now see this:

![Add a New Printer](image1)

In this example, we’ve connected an HP DeskJet F4200. Clicking on this printer will select it for installation, which, if successful, will display the following:

![Printing via Tails GUI](image2)

This indicates that Tails is attempting to install the USB printer. Assuming you receive no errors in this process, you will then see the following screen, which indicates that the printer is “ready” for printing.
32.2 Printing from the Command Line

You can also easily print from the command line using the `lp` command; if you haven’t already set your installed printer as default in the GUI, you can quickly do so by adding this line to your `~/.bashrc` file, or entering this directly into the terminal:

```
export PRINTER=Printer-Name-Here
```

If you need to find the name of the printer, you can use `lpstat` to get a list of installed printers, as such:

```
lpstat -a
```

Once you’ve set your default printer, you can easily print from the terminal by using the following syntax:

```
lp filename.extension
```

While printing from the GUI is much easier, once you’ve got everything set up, it’s equally straightforward from the command line, if you prefer that environment.
The SecureDrop Source Interface is served over a Tor Hidden Service, requiring an *.onion URL to access it. While Tor Onion Services provide end-to-end encryption by default, as well as strong anonymity, there are several reasons why you might want to consider deploying an additional layer of encryption and authentication via HTTPS:

- Extended Validation (EV) certificates, which are currently the only type of certificates that may be issued for *.onion addresses, are intended to attest to the identity of the organization running a service. This provides an additional measure of authenticity (in addition to the organization’s Landing Page and the SecureDrop Directory) to help assure sources that they are communicating with the intended organization when they access a given Source Interface.

- SecureDrop supports v3 onion services, which use updated cryptographic primitives that provide better transport-layer encryption than those used by v2 onion services. It is strongly recommended that you configure your instance to use v3 onion services, but if you cannot switch your instance to v3, using HTTPS on the source interface will provide an extra layer of encryption for data in transit.

### 33.1 Obtaining an HTTPS certificate for Onion URLs

DigiCert is currently the only Certificate Authority (CA) that issues HTTPS certificates for .onion sites. DigiCert requires organizations to follow the Extended Validation (EV) process in order to obtain a certificate for an Onion URL, so you should start by reviewing DigiCert’s documentation for obtaining a .onion certificate.

The EV certificates display in browsers with a green trust bar, including information about the organization:
The additional information about the organization, such as name and geographic location, are checked by the CA during the EV process. A Source can use this information to confirm the authenticity of a SecureDrop instance, beyond the verification already available in the SecureDrop Directory.

In order to obtain an HTTPS certificate for your SecureDrop instance, contact DigiCert directly. As part of the Extended Validation, you will be required both to confirm your affiliation with the organization, and to demonstrate control over the Onion URL for your Source Interface.

In order for you to demonstrate control over the Onion URL for your Source Interface, DigiCert will provide you with some text and ask you to make it available at a specific URL: `<onion_url>/.well-known/pki-validation/<unique_hash>.txt`. We have support for this workflow:

```
# From the Admin Workstation, SSH to the Application Server
$ ssh app
# Edit the validation file with content the CA provides
# Replace `<unique_hash>` with the token provided by Digicert
$ sudo vi /var/www/securedrop/.well-known/pki-validation/<unique_hash>.txt
```

**Note:** If you see “File Not Found” when navigating to this file in Tor Browser, check out the latest release in your `Admin Workstation` and re-run `./securedrop-admin install`.

While the CAB forum has specified that `.onion` certificates may have a maximum lifetime of 15 months, we have heard that some folks have run into issues with such certificates, and currently it seems safest to give the certificate a validity period of 12 months.

**Tip:** Be patient! HTTPS certificates for `.onions` are a recent and fairly niche development, so you may run into
various issues while trying to obtain the certificate.

**Warning:** As part of the process for obtaining an HTTPS certificate, you will need to generate a private key. This is usually stored in a file with a .key extension. It is **critical** that you protect this key from unauthorized access. We recommend doing this entire process on the Admin Workstation, and avoiding copying the .key to any insecure removable media or other computers.

### 33.2 Activating HTTPS in SecureDrop

Make sure you have *installed SecureDrop already.*

First, on the *Admin Workstation:*

```
cd ~/Persistent/securedrop
```

Make note of the Source Interface Onion URL. Now from ~/Persistent/securedrop on your admin workstation:

```
./securedrop-admin sdconfig
```

This command will prompt you for the following information:

| **Whether HTTPS should be enabled on Source Interface (requires EV cert):** | yes |
| **Local filepath to HTTPS certificate (optional, only if using HTTPS on source interface):** | sd.crt |
| **Local filepath to HTTPS certificate key (optional, only if using HTTPS on source interface):** | sd.key |
| **Local filepath to HTTPS certificate chain file (optional, only if using HTTPS on source interface):** | ca.crt |

The filenames should match the names of the files provided to you by DigiCert, and should be saved inside the *install_files/ansible-base/* directory. You’ll rerun the configuration scripts:

```
./securedrop-admin install
```

The webserver configuration will be updated to apply the HTTPS settings. Confirm that you can access the Source Interface at `https://<onion_url>`, and also that the HTTP URL `http://<onion_url>` redirects automatically to HTTPS.

**Note:** By default, Tor Browser will send an OCSP request to a Certificate Authority (CA) to check if the Source Interface certificate has been revoked. Fortunately, this occurs through Tor. However, this means that a CA or anyone along the path can learn the time that a Tor user visited the SecureDrop Source Interface. Future versions of SecureDrop will add OCSP stapling support to remove this request. See [OCSP discussion](#) for the full discussion.
Chapter 33. HTTPS on the Source Interface
Under a production installation post-install, the default way to gain SSH administrative access is over the Tor network. This provides a number of benefits:

- Allows remote administration outside of the local network.
- Provides anonymity to an administrator while logging into the SecureDrop servers.
- Can mitigate against an attacker on your local network attempting to exploit vulnerabilities against the SSH daemon.

Most administrators will need SSH access during the course of running a SecureDrop instance and a few times a year for maintenance. So the potential shortfalls of having SSH over Tor are not usually a major issue. The cons of having SSH over Tor can include:

- Slow and delayed remote terminal performance
- Allowing SSH access from outside of your local network can be seen as a potential larger security hole for some organizations, particularly those with tight network security controls.

That being said, the default setting of only allowing SSH over Tor is a good fit for most organizations. If you happen to require SSH restricted to the local network instead please continue to read.

### 34.1 Configuring SSH for Local Access

**Warning:** It is important that your firewall is configured adequately if you decide you need SSH over the local network. The install process locks down access as much as possible with net restrictions, SSH keys, and two-factor authentication. However, you could still leave the interface exposed to unintended users if you did not properly follow our network firewall guide.
**Warning:** This setting will lock you out of SSH access to your instance if your Admin Workstation passes through a NAT in order to get to the SecureDrop servers. If you are unsure whether this is the case, please consult your firewall configuration or network administrator.

**Note:** Whichever network you install from will be the one that SSH is restricted to post-install. This will come into play particularly if you have multiple network interfaces.

First, make sure your local SecureDrop environment is up-to-date and on the latest production release.

```bash
$ cd ~/Persistent/securedrop
$ ./securedrop-admin update
$ ./securedrop-admin setup
```

The setting that controls SSH over LAN access is set during the `sdconfig` step of the install. Below is an example of what the prompt will look like. You can answer either ‘no’ or ‘false’ when you are prompted for Enable SSH over Tor:

```bash
$ ./securedrop-admin sdconfig

Username for SSH access to the servers: vagrant
Local IPv4 address for the Application Server: 10.0.1.4
Local IPv4 address for the Monitor Server: 10.0.1.5
Hostname for Application Server: app
Hostname for Monitor Server: mon

[...]
Enable SSH over Tor (recommended, disables SSH over LAN). If you respond no, SSH will only be available over LAN only: no
```

Then you’ll have to run the installation script:

```bash
$ ./securedrop-admin install
```

**Note:** If you are migrating from a production install previously configured with SSH over Tor, you will be prompted to re-run the `install` portion twice. This is due to the behind the scenes configuration changes being done to switch between Tor and the local network.

Finally, re-configure your Admin Workstation as follows:

```bash
$ ./securedrop-admin tailsconfig
```

Assuming everything is working you should be able to gain SSH access as follows:

```bash
$ ssh app
$ ssh mon
```
SecureDrop On-Site Training Schedule

This is a high level schedule for what happens for the 2 days during an on-site install.

35.1 Day 1: Preparation and Install

35.1.1 Setup and Introductions

Time: 30min
Participants: all
Required: projector, WiFi access, pre-configured demo SecureDrop instance and 2 laptops to act as the Journalist Workstation and SVS

- The demo instance has multiple sources to try and give a feel of what it will look like at 2 weeks past being public with sources in different states of the reply process

35.1.2 Overview of SecureDrop

Time: 2 hours
Participants: journalists, editors, SecureDrop admins, OSSEC alert recipients and anyone else interested

- Go over the SecureDrop FAQs
- Go over the SecureDrop environment diagrams
- Importance of the Landing Page security and Twitter feedback
- Demo the source submission process
- Demo the journalist’s processes for checking the Journalist Interface
- Demo the journalist’s processes for replies
- Demo working with submissions on the SVS
• Discuss scrubbing submitted documents prior to publication
• Options for distributing with other news organizations
• Show example of an OSSEC alert, briefly cover what it does
• Show example of ‘is it up?’ Nagios monitoring alerts for Source Interface
• Explain why the Journalist Interface does not have ‘is it up?’ monitoring
• Discuss vanity onion URLs with Shallot and Scallion
• How to brand the Source Interface and Journalist Interface
• Physical security of servers and SVS
• How to securely publicize the organization’s Source Interface Tor URL
• Distribute important info:
  – Third-party security mailing lists to subscribe to
  – https://freedom.press/about/staff
  – https://securedrop.org
  – https://docs.securedrop.org
  – Hardware for SecureDrop
  – Overview guidelines
  – Source Best Practice Guide
  – Journalist Best Practice Guide
  – Admin Best Practice Guide
• Answering the client vs. server side crypto debate
• Link to security audits

35.1.3 Questions

Time: 30 min

35.1.4 Installing SecureDrop

Time: 6 hours

  • Follow Installing SecureDrop

35.2 Day 2: Journalist and Admin Training

35.2.1 Journalist Training

Time: 2 separate sessions, about 2 hours each
Participants: journalists and admins

  • Check access to previously created Tails USB
• Generate personnel GPG keys
• Setup KeyPassX manager (one for SVS, one for personnel Tails)
• Options between YubiKey/FreeOTP app for 2FA (SSH, Journalist Interface, FDE and password managers)
• Secure-deleting and difference between wipe and erase free space on Tails, and when to use each
• Disaster recovery for 2FA and password manager, personnel GPG keys
• Updating Tails
• Backing up the SVS
• If needed, process for distributing the Application’s private GPG key to a distant journalist’s air-gapped SVS
• Do complete journalist process walk through twice, either on different days or between morning/afternoon sessions
• Using MAT (Metadata Anonymisation Toolkit)
• What to do for unsupported formats

35.2.2 Admin Training

Time: 2 hours
Participants: admins

• Check access to previously created Tails USB
• Generate personnel GPG keys
• Setup KeyPassX manager (one for SVS, one for personnel Tails)
• Options between YubiKey/FreeOTP app for 2FA (SSH, Journalist Interface, FDE and password managers)
• Secure-deleting and difference between wipe and erase free space on Tails, and when to use each
• Disaster recovery for 2FA and password manager, personnel GPG keys
• Updating Tails
• Setting up SSH aliases for the Admin Workstation
• How to use screen or tmux to help prevent being locked out of the system
• Adding packages to Tails
• Go over common OSSEC alerts for security updates and daily reports
• Disaster recovery for application, remote access and SVS
• Common admin actions
• Adding/removing users
• Enabling logging
• Sending logs to FPF
• Generating new Tor Onion Services
• Updating application’s GPG key
• Re-IP’ing
• Backups
• Disk space monitoring
• Updating SMTP and OSSEC alert configs
• Changing passphrases (for FDE, persistent volumes, 2FA, KeePassX managers . . .)
• What will happen to local modifications to prod system after updates
• Updating SecureDrop Application
  – Unattended upgrades
  – Upgrades that require admin intervention
This guide describes in detail how to set up a YubiKey for two-factor authentication on the Journalist Interface. This setup is performed once per journalist to create a secure log-in method. The process requires some configuration steps using a separate software tool.

**Note:** You will do all of these steps from within the Tails operating system.

### 36.1 What is a YubiKey?

A YubiKey is a physical token used for two-factor authentication. They are made by a company called Yubico and are commercially available.

### 36.2 Download and Launch the YubiKey Personalization Tool

1. Start Tails. At the log in-screen, choose the option to allow an administrator passphrase.
2. Open a terminal and enter

   ```bash
   sudo apt-get update;
sudo apt-get install yubikey-personalization-gui
   ```

1. Once you have downloaded and installed the personalization program, open a **Root Terminal** by choosing **Applications** > **System Tools** > **Root Terminal**.
2. Open the YubiKey personalization tool by entering

   ```bash
   yubikey-personalization-gui
   ```
36.3 Setting Up Hardware-Based Codes

After opening the personalization tool, click the heading OATH-HOTP. This will bring you to a window called Program in OATH-HOTP mode.

Click on the Quick button.

Under Configuration Slot, click Configuration Slot 1.

**Note:** If you are already using this YubiKey for something else, you should choose Configuration Slot 2. You will have to press and hold for several seconds to use the token from Slot 2 instead of the one in Slot 1. See the YubiKey manual for more information.

In the section titled OATH-HOTP parameters, uncheck the box for OATH Token Identifier (6 bytes). Next, uncheck the box for Hide secret. This will display the Secret Key (20 bytes Hex) field.

**Important:** Make a note somewhere safe of the Secret Key (20 bytes Hex) value.
When ready, click the **Write Configuration** button.

Click through the warning about overwriting the configuration slot and choose a location to save the log file. When the configuration is done, you should see green text saying **YubiKey configured** at the top of the window.

### 36.4 Adding Users

When adding new users, a SecureDrop admin will need the **Secret Key** value described above. She will enter it after selecting the **I’m Using a YubiKey** option while *adding users*.

### 36.5 Using Your YubiKey

When using a Yubikey to log-in to the *Journalist Interface*, insert the Yubikey into the USB port and enter your username and passphrase. Then click the **Two-factor Code** field to focus the cursor there. Quickly press the lighted button on your YubiKey. This will insert the 6-digit code that you will need to log in.
Note: When using Configuration Slot 2, be sure to press and hold the YubiKey button for approximately 3 seconds.
There are a number of reasons why you might want to backup and restore a SecureDrop installation. Maintaining periodic backups is generally a good practice to guard against data loss. In the event of hardware failure on the SecureDrop servers, having a recent backup will enable you to redeploy the system without changing Onion URLs, recreating journalist accounts, or losing previous submissions from sources.

Note: Only the Application Server is backed up and restored, including historical submissions and both Source Interface and Journalist Interface URLs. The Monitor Server needs to be configured from scratch in the event of a hardware migration.

37.1 Minimizing Disk Use

Since the backup and restore operations both involve transferring all of your SecureDrop’s stored submissions over Tor, the process can take a long time. To save time and improve reliability for the transfers, take a moment to clean up older submissions in the Journalist Interface. As a general practice, you should encourage Journalists to delete regularly unneeded submissions from the Journalist Interface.

Tip: Although it varies, the average throughput of a Tor Hidden Service is about 150 kB/s, or roughly 4 hours for 2GB. Plan your backup and restore accordingly.

You can use the following command to determine the volume of submissions currently on the Application Server: log in over SSH and run `sudo du -sh /var/lib/securedrop/store`.

Note: Submissions are deleted asynchronously and one at a time, so if you delete a lot of submissions through the Journalist Interface, it may take a while for all of the submissions to actually be deleted. SecureDrop uses `shred` to securely erase files, which takes significantly more time than normal file deletion. You can monitor the progress of queued deletion jobs by logging in to the Application Server over SSH and running:
If you find you cannot perform a backup or restore due to this constraint, and have already deleted old submissions from the Journalist Interface, contact us through the SecureDrop Support Portal.

### 37.2 Backing Up

Open a Terminal on the Admin Workstation and cd to your clone of the SecureDrop git repository (usually ~/.Persistent/securedrop). Ensure you have a tagged SecureDrop release checked out, version 0.4 or later. (You can run `git describe --exact-match` to verify that you have the right source checked out.)

**Note:** The backups are stored in the Admin Workstation persistent volume. Verify that you have enough space to store the backups before running the backup command.

You can use the `du` command described earlier to get the approximate size of the backup file (since the majority of the backup archive is the stored submissions), and Tails’ Disks utility to see how much free space you have on your persistent volume.

#### 37.2.1 Check Connectivity

First, verify that your Admin Workstation is able to run Ansible and connect to the SecureDrop servers.

```
ssh app uptime
```

If this command fails (usually with an error like “SSH Error: data could not be sent to the remote host. Make sure this host can be reached over ssh”), you need to debug your connectivity before proceeding further. Make sure:

- Ansible is installed
- the Admin Workstation is connected to the Internet
- Tor starts successfully
- The appropriate onion service configuration files are present in ~/.Persistent/securedrop/install_files/ansible-base and the `./securedrop-admin tailsconfig` command completes successfully

If Ansible is not installed, or `./securedrop-admin tailsconfig` fails, see [Configure the Admin Workstation Post-Install and Create Backups](#) for detailed setup instructions.

#### 37.2.2 Create the Backup

When you are ready to begin the backup, run

```
./securedrop-admin backup
```

The backup action will display itemized progress as the backup is created. Run time will vary depending on connectivity and the number of submissions saved on the Application Server.

When the backup action is complete, the backup will be stored as a compressed archive in `install_files/ansible-base`. The filename will begin `sd-backup` followed by a timestamp of when the backup was initiated, and end with `.tar.gz`. You can find the full path to the backup archive in the output of backup action.
Warning: The backup file contains sensitive information! It should only be stored on the Admin Workstation, or on a dedicated encrypted backup USB.

Note: When dealing with larger backups, the securedrop-admin backup command may fail with a MemoryError at this stage of the operation: “Fetch the backup tarball back to the Admin Workstation”.

If this happens, a backup was successfully generated, but it is still on the server. Run this command from your ~/Persistent/securedrop directory to copy the backup your Admin Workstation:

```
rsync -av --progress --partial app:$(ssh app ls -lrt /tmp/sd-backup* | tail -1) → install_files/ansible-base/
```

If the transfer fails or is interrupted, you can simply run this command again to resume it.

Note that this method will only work if you have first run the securedrop-admin backup command, and the backup has successfully progressed at least until the “Fetch the backup tarball” stage.

### 37.3 Restoring

#### 37.3.1 Prerequisites

The process for restoring a backup is very similar to the process of creating one. As before, boot the Admin Workstation and cd to the SecureDrop repository. Ensure that you have SecureDrop 0.4 or later checked out.

The restore command expects to find a .tar.gz backup archive in install_files/ansible-base under the SecureDrop repository root directory. If you are using the same Admin Workstation to do a restore from a previous backup, it should already be there because it was placed there by the backup command. Otherwise, you should copy the backup archive that you wish to restore to install_files/ansible-base.

Note: The backup strategy used for SecureDrop versions prior to 0.3.7 created encrypted archives with the extension .zip.gpg. You can safely remove those files once you've created the .tar.gz backup archive described in this guide.

#### 37.3.2 Restoring From a Backup File

**Important:** This documentation applies to a SecureDrop instance using v2 onion services. If your instance uses v3 onion services, you will need to follow additional steps depending on your specific restore scenario.

To perform a restore, you must already have a backup archive. Provide its filename in the following command:

```
./securedrop-admin restore sd-backup-2017-07-22--01-06-25.tar.gz
```

Make sure to replace sd-backup-2017-07-22--01-06-25.tar.gz with the filename for your backup archive. The backup archives are located in install_files/ansible-base.

Once the restore is done, the Application Server will use the original Source Interface and Journalist Interface Onion URLs. You will need to update the corresponding files on the Admin Workstation:
Once ssh access to the servers has been established (or if using ssh over local network), Onion URLs for the Source Interface and Journalist Interfaces can be fetched using the installer:

```
./securedrop-admin install
```

Then rerun `./securedrop-admin tailsconfig` to update the Admin Workstation to use the restored Onion URLs again. See *Configure the Admin Workstation Post-Install and Create Backups* for detailed instructions.

### 37.4 Migrating

Moving a SecureDrop installation to new hardware consists of

1. *Backing up* the existing installation;
2. *Installing* the same version of SecureDrop on the new hardware;
3. *Restoring* the backup to the new installation.
38.1 Backup the Workstations

Note: This workflow will create a single USB drive with the data backed up from all Tails drives. If instead you’d like to create a single duplicate Tails drive, you should follow the official documentation maintained by the Tails project.

Now that you have set up the Secure Viewing Station, the Admin Workstation, and your Journalist Workstations, it is important you make a backup. Your USB drive may wear out, a journalist might lose their drive, or something completely unexpected may happen.

In all these cases, it is useful to have a backup of your data for each device.

38.1.1 What You Need

1. You will need your existing SecureDrop Tails USB sticks (Admin Workstation, Journalist Workstation, and Secure Viewing Station).
2. You will also need an airgapped machine to perform the backups. The Secure Viewing Station may be used for this task.
3. You will also need a “primary” Tails USB, which we will use to perform the backups.
4. You also need at least one USB drive to backup the data from your current SecureDrop Tails USB sticks.

Warning: An airgapped machine (such as the Secure Viewing Station) is required in order to perform these backups safely. By isolating the machine from all network access, you reduce the exposure of sensitive data to networked computers, thereby reducing the threat of compromise by adversaries who wish to gain access to your SecureDrop instance.
The airgapped machine should have 3 USB ports, so you can plug in the primary Tails USB drive, the Tails drive you want to backup, and the backup drive at the same time. If you don’t have 3 USB ports available you can use a USB hub which may reduce transfer speeds.

Note: The steps in this section should be performed for each Secure Viewing Station, Journalist Workstation, and Admin Workstation USB drive in your organization.

### 38.1.2 Preparing the Backup Device

First you must boot the primary Tails USB drive. Ensure you set an administrator password set at the login screen. Then navigate to Applications Utilities Disks.
Insert the USB drive you wish to use as a backup drive.

Select the drive from the list of drives in the left column.

Click the button with the two cogs and click **Format Partition**...
Fill out the form as follows:
• **Erase**: Don't overwrite existing data (*Quick*)
• **Type**: Encrypted, compatible with Linux systems (*LUKS + Ext4*)
• **Name**: Backup

**Warning**: Since this will serve as a long-term backup, make sure to use a strong passphrase.

Click **Format**.

A dialog box will appear asking you **Are you sure you want to format the volume?** appears, click **Format**.

Once completed, you will see two partitions appear:
Now that you made the backup device, plug in the device you want to backup. Then, browse to **Places Computer:**
Click on the disk on the left side column. Fill in the passphrase you set up when you *created your Tails devices*.
You should now have both the Backup and TailsData partition to be backed up mounted and ready to access.
38.1.3 Create the backup using Rsync

Open a terminal by going to Applications Favorites Terminal.

Next, create a directory on the Backup USB for the device to be backed up - the command below creates a directory named admin-backup:

```
mkdir /media/amnesia/Backup/admin-backup
```

Then, copy the contents of the device’s persistent volume to the directory using rsync:

```
sudo bash -c "rsync -a --info=progress2 --no-specials --no-devices /media/amnesia/TailsData/ /media/amnesia/Backup/admin-backup/ && sync"
```

**Note:** Please make sure to include the trailing / in the directory paths in the command above, otherwise the files will not be backed up correctly.

Once complete, unmount the TailsData partition by clicking the Eject button beside its entry in the lefthand column of the file manager. When its entry is no longer shown in the lefthand column, it is safe to remove the Admin Workstation USB.

Repeat these steps for every device, making a new folder on the backup device for each device you back up.

Finally, once you have completed the steps described in this section for each USB drive, unmount the Backup partition by clicking its Eject button. Wait until the Backup USB can be safely removed, and store it somewhere safely.
38.2 Restoring a Workstation from a Backup

To recreate a backed-up Admin Workstation, Journalist Workstation, or Secure Viewing Station Tails USB, you will need

- your Backup USB containing the persistent volume to be restored,
- a blank USB stick to be set up as the new workstation USB,
- an airgapped machine and a USB with Tails already installed, referred to as the host Tails USB in this document. The host Tails USB is only used to transfer files between the Backup USB and the new workstation USB.

The process will require 3 USB ports - if necessary, you can use a USB hub. We recommend labeling USB devices before use, as it can be easy to confuse them.

38.2.1 Prepare the new Tails USB

Follow the guide to creating a Tails USB to install Tails and create a persistent volume on the blank USB stick to create the new workstation USB.

38.2.2 Open the Backup USB and new Tails Persistent Volume

First, boot up the host Tails USB on the airgapped machine, making sure to set an administration password on the Tails Greeter dialog.

Then, navigate to Places Computer to open the file manager, and insert the Backup USB. Click its entry in the lefthand column and enter its decryption passphrase when prompted. Its volume name (Backup in the instructions above) will appear in place of the generic N.M GB Encrypted name.

Next, insert the new workstation USB, and click its entry in the lefthand column. When prompted, enter its persistent volume’s passphrase. The volume name TailsData will appear in the lefthand column.

38.2.3 Copy the Backup to the New Workstation USB’s Persistent Volume

Open a terminal by navigating to Applications Favorites Terminal. Next, use the rsync command to copy the appropriate backup folder to the new workstation USB’s persistent volume. For example, if the backup folder to be copied is named admin-backup, run the following command:

```bash
sudo bash -c "rsync -a --info=progress2 --no-specials --no-devices /media/amnesia/Backup/admin-backup/ /media/amnesia/TailsData/ && sync"
```

Note: Please make sure to include the trailing / in the directory paths in the command above, otherwise the backup files will not be restored correctly.

Once the command is complete, click the Eject button for the TailsData volume in the lefthand column of the file manager, wait for the TailsData entry to disappear from the column, and remove the new workstation USB.
You may now repeat the restore process for any other USBs that you wish to restore, or shut down the host Tails USB and test your new workstation USB by booting it with persistence unlocked and verifying its functionality.
Updating Tails USBs

You are responsible for keeping your Tails USBs updated.

There are two ways to perform updates: via the Tails graphical installer and manually. The manual update process will work on any Tails USB. The graphical installer requires an Internet connection to notify you when updates are available, so it is only suitable for internet-connected Tails workstations (such as the Admin Workstation and the Journalist Workstation).

Because the Secure Viewing Station (SVS) is airgapped, it cannot receive update notifications, so it will need to be updated manually.

**Important:** The SVS stores some of SecureDrop’s most sensitive data (decrypted submissions, the Submission Private Key), so we **strongly** encourage you to manually update the SVS whenever a new version of Tails is released.

The Tails project releases updates every 6 to 8 weeks. Occasionally they release a new version ahead of the normal cycle in order to address a security issue. For regular Tails OS and security information, check out the Tails Security page and subscribe to the Tails RSS/Atom feed.

### 39.1 Update via Graphical Installer

For Internet-connected Tails workstations, you’ll see a notification when a new version of Tails is available.
We recommend that you back up your existing configuration and update as soon as possible. It usually takes some time for updates to download, so keep that in mind when choosing when to update.

### 39.2 Update Manually

The process for manually updating a Tails USB is:

1. **Recommended:** Make a backup of the USB you want to update.
2. Install the latest version of Tails on your primary Tails USB or a blank USB stick.
3. Use the primary Tails USB to perform a manual update of your desired Tails USB on a fully offline (airgapped) computer, such as the computer used for the Secure Viewing Station.

#### 39.2.1 What you need

1. Your Admin Workstation computer
2. A primary Tails USB stick (you may still have one; it was used to create the Admin Workstation, Secure Viewing Station, and Journalist Workstation Tails USBs during the initial SecureDrop install process)
3. The Tails USB that you want to update
4. A Backup USB to back up the data on your existing Tails USBs
5. An airgapped computer, such as the computer used for the Secure Viewing Station.
39.2.2 1. Back up your Tails USB

Follow the instructions to back up your existing Tails USB.

39.2.3 2. Get the latest version of Tails

If you have an existing primary Tails USB, boot into it on your Admin Workstation computer and follow the graphical updater prompts that guide you through the update process.

Alternatively, you can also download and install the newest version of Tails from scratch (as you did when you first installed SecureDrop). This may be faster if your primary Tails USB has not been updated in a while.

39.2.4 3. Perform airgapped update

In this step, use the up-to-date primary Tails USB to update your desired Tails USB.

**Warning:** The entire Secure Viewing Station is designed to be airgapped, so the SVS Tails USB must never be plugged into a computer with a network connection.

Use an airgapped computer, such as the Secure Viewing Station computer, to perform the steps in this section.

Plug your primary Tails USB into the airgapped computer and boot into Tails.

You can then perform the manual update steps.

39.2.5 If you encounter issues

If you run into issues, you can always restore your data from the backup device following the instructions here.

If you continue to have problems, you can contact us through the SecureDrop Support Portal.
Note: These instructions refer to a SecureDrop instance using v2 onion services. If your instance uses v3 onion services and you need to rebuild your Admin Workstation, please contact FPF through the SecureDrop Support Portal.

In cases where an Admin Workstation USB stick has been lost or destroyed, and no backup exists, it is possible to rebuild one. In order to do so, you’ll need

- physical access to the SecureDrop servers
- 2 USB sticks:
  - Tails Master USB
  - 1 replacement Admin Workstation USB (USB3 and 16GB or better recommended)

The process requires experience with the Linux command line and Tails, and can take up to 3 hours. If a backup of the SecureDrop application server is available, reinstalling the instance may be simpler. An outline of the steps involved in rebuilding an Admin Workstation is as follows:

1. Prepare the USB sticks.
2. (Optional) Boot the Application and Monitor Server in single user mode and reset the shell admin account password.
3. Set up SSH access for the new Admin Workstation.
4. Retrieve SecureDrop configuration settings from the Application and Monitor Server.
5. Back up and configure the SecureDrop application.
6. Run the ./securedrop-admin tailsconfig and ./securedrop-admin install commands from the new Admin Workstation.
7. Complete post-rebuild tasks.

Important: The rebuild process involves temporarily removing iptables rules on the Application and Monitor Servers, weakening their security. Because of this, it’s important to complete the rebuild process promptly, to avoid
leaving the servers in an insecure state.

40.1 Step 1: Prepare the USB sticks

First, create a new Admin Workstation USB and set up a persistent volume with a strong passphrase.

Once persistence has been set up, start up the Admin Workstation with persistence enabled, install the SecureDrop application code, and set up the KeePassX database.

The Admin Workstation uses SSH with key authentication to connect to the servers, so you’ll need to create a new SSH keypair for your SecureDrop instance. To do so, open a terminal by navigating to Applications>Favorites>Terminal, and run the following command:

```bash
ssh-keygen -t rsa -b 4096
```

When prompted to Enter file in which to save the key, Press Enter to use the default location. When prompted for a passphrase, it’s safe to leave it blank.

40.2 Step 2: (Optional) Boot the servers in single-user mode

If you do not have the original password for the shell admin account on the Application and Monitor Servers, you’ll need to reset the password on each server by booting in single user mode. In order to do so, you’ll need physical access to the server, a keyboard, and a monitor.

First, connect a monitor and keyboard to the Monitor Server. Then reboot the server. When the GRUB menu appears, make sure the Ubuntu entry is highlighted, and press e to edit boot options. In the boot options for Ubuntu, find the line that starts with linux and ends with ro. Add single after ro, separated by a space, and press F10 to boot in single user mode.

40.2.1 Reset the SecureDrop admin user’s password

Once the root prompt appears, you’ll need to reset the password for the SecureDrop admin user. By default this user is named sdadmin and has UID 1000. However it may have been named differently during the installation of your instance. You can use the command getent passwd 1000 to check the username corresponding to UID 1000. Once you have the correct username, reset its password using the passwd command, for example:

```bash
passwd sdadmin
```

Important: Make sure to select a strong password, and record it in the Admin Workstation’s KeePassX database.

Finally, reboot the Monitor Server and verify that you can log in at the console using the new password.

Repeat the process for the Application Server. Use the same username and password as for the Monitor Server - this is required in order for the ./securedrop-admin install command to work correctly.

40.3 Step 3: Set up Admin Workstation access

Next, you’ll configure the servers to allow temporary SSH access from the new Admin Workstation.
First, start the new Admin Workstation with persistence enabled and an administration password set.

Next, connect the new Admin Workstation to the Hardware Firewall via the appropriate Ethernet port, and set up its static IP address. For more information on how to do so, see this section in the firewall setup documentation. If you do not know the correct static IP address for the Admin Workstation, and you are using a recommended pfSense-based Hardware Firewall, you can retrieve the address by logging into its admin interface and checking the settings under Firewall Aliases.

Note: If you do not have login credentials for your pfSense firewall, check its user manual for instructions on resetting the administration password.

Next, determine whether your instance was set up to allow administrative access via SSH over Tor, or via SSH over LAN. If you don’t know which option was originally chosen, you can check as follows:

1. Log in to the Application Server via the console using the administration username and password.
2. Check to see if an SSH hidden proxy service exists, using the command `sudo cat /var/lib/tor/services/ssh/hostname`. If this file exists and includes an Onion URL and authorization token, your instance is set up to use SSH over Tor and you should configure temporary SSH access using these instructions. If not, your instance is set up to use SSH over LAN, and you should follow these instructions instead.

**40.3.1 Configuring access for an SSH-over-Tor instance**

Direct SSH access is disabled when the SSH-over-Tor option is selected during installation. To temporarily re-enable it, you’ll need to update iptables rules and change the sshd daemon’s configuration.

First, log on to the Application Server via the console, and run the following commands, substituting the Admin Workstation’s static IP for `<admin_static_ip>`:

```bash
sudo iptables -I INPUT -p tcp --dport 22 -s <admin_static_ip> -m state --state NEW,ESTABLISHED -j ACCEPT
sudo iptables -I OUTPUT -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

Next, edit the file `/etc/ssh/sshd_config`, changing the line:

```
ListenAddress 127.0.0.1:22
```

to:

```
ListenAddress 0.0.0.0:22
```

and deleting the line:

```
PasswordAuthentication no
```

Then, restart sshd using the command `sudo service sshd restart`.

Finally, check the file `/etc/network/interfaces`, and note down the network settings for the default Ethernet interface. You’ll need them in the next step.

Repeat the process above for the Monitor Server, making sure to note down its network settings as well.

Once the Monitor Server has been configured, proceed to enable access from the new Admin Workstation.

---

**40.3. Step 3: Set up Admin Workstation access**
40.3.2 Configuring access for an SSH-over-LAN instance

First, log on to the Application Server via the console and edit the file /etc/ssh/sshd_config, deleting the line:

```
PasswordAuthentication no
```

Then, restart sshd using the command `sudo service sshd restart`.

Finally, check the file /etc/network/interfaces, and note down the network settings for the default Ethernet interface. You’ll need them in the next step.

Repeat the process above for the Monitor Server, making sure to note down its network settings as well.

40.3.3 Enabling access from the new Admin Workstation

From the Admin Workstation, open a terminal and copy the Admin Workstation’s SSH public key to the servers, substituting the values for the server administration username and server IP addresses in the commands below and entering the admin account’s password when prompted:

```
ssh-copy-id <admin-username>@<application-server-ip>
ssh-copy-id <admin-username>@<monitor-server-ip>
```

Next, create a file `~/.ssh/config` with contents as below, again substituting the appropriate values for your servers:

```
Host app
  User <admin-username>
  Hostname <application-server-ip>
  ProxyCommand none

Host mon
  User <admin-username>
  Hostname <monitor-server-ip>
  ProxyCommand none
```

Finally, test direct SSH access from the terminal, using the commands `ssh app` and `ssh mon`. It should be possible to connect without entering a password.

40.4 Step 4: Retrieve SecureDrop configuration info from the servers

In addition to the account and networking information retrieved from the servers so far, you’ll need to retrieve the following files and info:

- Tor Onion Service URLs and tokens
- GPG Submission Public Key and OSSEC public key
- OSSEC alert configuration details
- (Optional) HTTPS configuration details

40.4.1 Retrieve Onion Service info

In order to connect to the Tor Onion Services used by your instance, you will need to copy their details to the Admin Workstation. To do so, run the following commands from a Terminal window:
If your instance was set up to use SSH over TOR, you’ll also need to copy over the details of the SSH proxy Onion Services, by running the following commands:

```bash
echo "HidServAuth $(ssh app sudo cat /var/lib/tor/services/journalist/hostname)" > app-journalist-aths
echo "$(ssh app sudo cat /var/lib/tor/services/source/hostname)" > app-source-ths
```

40.4.2 Retrieve GPG Public Keys

There are 2 GPG public keys required by the instance configuration, that you will need to copy to the new Admin Workstation.

To copy the Submission Public Key, download it from the Source Interface and import it locally using the following commands:

```bash
cd ~/Persistent/securedrop/install_files/ansible-base
curl http://$(cat app-source-ths)/journalist-key > SecureDrop.asc
gpg --import SecureDrop.asc
```

Validate that the imported key’s fingerprint matches the one on your SecureDrop install. You can do this by first running the command:

```
gpg --with-fingerprint --import-options import-show --dry-run --import SecureDrop.asc
```

Then, compare the returned fingerprint value with that advertised by your Source Interface, using the command:

```bash
curl http://$(cat app-source-ths)/metadata
```

To copy the OSSEC alert public key, first list available keys on the monitor server:

```
ssh mon sudo gpg --homedir=/var/ossec/.gnupg --k
```

Look for the key corresponding to the destination email address for OSSEC alerts. Then, import it locally using the following commands (substituting the appropriate email address for alerts@example.com):

```
ssh mon sudo gpg --homedir=/var/ossec/.gnupg --export --armor alerts@example.com > ossec.pub
gpg --import ossec.pub
```

You will be prompted for the fingerprints for both keys during the next step. To view their fingerprints, use the command:

```
gpg -k --fingerprint
```

40.4.3 Retrieve OSSEC alert configuration details

In addition to the OSSEC alert GPG key you retrieved above, you’ll also need the following configuration information:

- SMTP server
To retrieve these values, use the following command in the terminal:

```bash
ssh mon sudo cat /etc/postfix/sasl_passwd
```

This will return a line like:

```
[smtp.gmail.com]:587 testossec@gmail.com:AwfulPassword
```

In this example, smtp.gmail.com is the SMTP server, 587 is the SMTP port, testossec is the SASL username, gmail.com is the SASL domain, and AwfulPassword is the SASL password.

### 40.4.4 (Optional) Retrieve HTTPS certificate files

If your *Source Interface* was configured to use HTTPS, you will need to copy three related files from the *Application Server* to the *Admin Workstation*.

To retrieve these files, use the commands:

```bash
cd ~/Persistent/securedrop/install_files/ansible-base
ssh app sudo tar -c -C /var/lib ssl/ | tar xvf -
```

These commands will create a directory named `~/Persistent/securedrop/install_files/ansible-base/ssl` on the *Admin Workstation*, containing your instance’s SSL certificate, certificate key, and chain file. When prompted for the names of these files during the next step, you should specify them relative to the `install_files/ansible-base` directory, i.e. as `ssl/mydomain.crt`.

### 40.5 Step 5: Configure and back up the application

Next, configure the application using the files and info retrieved in the previous steps. To do so, connect to the Tor network on the *Admin Workstation*, open a Terminal and run the following commands:

```bash
cd ~/Persistent/securedrop
./securedrop-admin setup
./securedrop-admin sdconfig
```

**Note:** The `./securedrop-admin setup` command may take several minutes to complete, and may fail due to network issues. If it fails, it’s safe to run again.

When prompted, fill in the values retrieved in the previous steps.

Once complete, configure the *Admin Workstation* using the command:

```
./securedrop-admin tailsconfig
```

This will set up desktop shortcuts for the *Source* and *Journalist Interfaces*, and configure administrative access to the servers.
Once the ./securedrop-admin tailsconfig command is complete:

- verify that the Hostname references in ~/.ssh/config have been updated to refer to Onion URLs instead of direct IP addresses,
- verify that you can connect to the servers using ssh app and ssh mon, accepting the host verification prompt if necessary,
- and verify that the desktop shortcuts for the Source and Journalist Interfaces work correctly, opening their respective homepages in the Tor Browser.

Next, back up the servers by running the following command in the terminal:

```
./securedrop-admin backup
```

### 40.6 Step 6: Run the ./securedrop-admin install command

After the ./securedrop-admin backup command completes successfully, you should undo the changes made to enable temporary local SSH access, by running the following command:

```
./securedrop-admin install
```

Once the command completes successfully, your new Admin Workstation is ready for use. To revert the changes made to enable temporary local SSH access, you should reboot the servers, by issuing the following commands in a terminal:

```
ssh app sudo reboot
ssh mon sudo reboot
```

### 40.7 Step 7: Post-rebuild tasks

We recommend completing the following tasks after the rebuild:

- Set up a new administration account on the Journalist Interface, by following these instructions
- Verify that submissions can be decrypted, by going through the decryption workflow with a new submission.
- Back up your Admin Workstation using the process documented here.
- Delete invalid admin accounts in the Journalist Interface.
- Restrict SSH access to the Application and Monitor Servers to valid Admin Workstations. If your new Admin Workstation USB stick is the only one that should have SSH access to the servers, you can remove access for any previous Admin Workstations from the terminal, using the commands:

```
cd ~/Persistent/securedrop
./securedrop-admin reset_admin_access
```

You can also selectively remove invalid keys by logging on to the Application and Monitor Servers and editing the file ~/.ssh/authorized_keys, making sure not to remove the public key belonging to your new Admin Workstation.

- Optionally, set up daily journalist alerts, by running ./securedrop-admin sdconfig and providing a valid GPG key and fingerprint, along with the corresponding destination email address, then running ./securedrop-admin install again to update the server configuration.
Kernel updates address known bugs and security vulnerabilities in the Linux kernel. They may be installed automatically on your Application and Monitor Servers as part of a SecureDrop release. All kernel updates are tested extensively against recommended hardware. If things do go wrong (e.g., the server does not boot after a kernel update), the following instructions will help you to roll back to the previous, working kernel. You can then report compatibility issues to us so we can work together to resolve them as quickly as possible.

First, you need to physically access each server. Power down the server (safely if possible), attach required peripherals (keyboard, monitor), and power the server back up.

If you have access to the password for your admin user, you can use it to log into each server without the use of two-factor authentication, which was disabled for keyboard logins in SecureDrop 0.8.0. You may have saved the password in the KeePassX database on your Admin Workstation. If you do not have the password, you can boot into single user mode instead.

### 41.1 Boot into Single User Mode

To access single user mode, you will have to edit the boot options for the new kernel. You can do so using the GRUB bootloader, pictured below:
Press any key quickly just once. You will only have about 2 to 3 seconds before Ubuntu starts booting. If you miss that window, just log in normally and reboot safely, provided you can log in. Do not unplug or forcibly shut down the server.

Once you hit a key, you will be able to interact with the menu with the up () and down () keys. Select “Ubuntu” as shown above, and press “e” to edit the boot options. In the line that begins with “linux”, add the word “single” at the end. When you are done, the output on your console should look similar to the screenshot below.
41.2 Test the New Kernel

Observe the boot process. It is possible that the system will fail to boot completely; if so, the log information will help us to understand what is happening.

Provided that you can log in, check if you have network access. Try a command such as `sudo host freedom`. If you don’t have network access, it is most likely due to the upgraded kernel missing a network driver for your hardware.

If everything appears to be operating normally, the outage may not be kernel-related. In that case, you may still wish to follow the steps at the end of this document to send us log information along with an issue report, and we will help you investigate.

If you are experiencing network issues or other kernel problems, we recommend that you roll back to an older kernel, and that you report the issue to us immediately.

41.3 Compare the Behavior of the Old Kernel

Reboot the server in a safe way with `sudo reboot`. After the BIOS screen, you can select a different kernel from the GRUB boot menu by selecting **Advanced options for Ubuntu**, pictured below.
The next menu should give you a list of kernels, similar to the one pictured below:
Choose the option with the previous kernel version. If unsure, please consult the release notes for the most recent release of SecureDrop, which will include details about kernel version changes.

As before, you may need to edit the kernel options to enter single user mode. The boot process should proceed normally. Wait until you get a login prompt and log in.

Once you are logged in, check to see if you have network access. If you do, then your instance is having an issue with the newer kernel. In that case, we need to temporarily set an older kernel as the default.

### 41.4 Roll Back to the Old Kernel

**Important:** It is of critical importance for the security of your instance that we work together to resolve any compatibility issues. Rolling back to an older version is only a stopgap measure to avoid a prolonged outage of your SecureDrop instance.

Inspect the file `/boot/grub/grub.cfg`. You should find a `menuentry` line with the same text that you selected during boot, e.g.:

```
submenu 'Advanced options for Ubuntu'...
   menuentry 'Ubuntu, with Linux 4.xxx.xx-grsec...
```

Take note of its position among the other submenu entries (it will most likely be third). Then edit the GRUB configuration:
sudo nano /etc/default/grub

Make a backup of the file or take a note of the current value of GRUB_DEFAULT somewhere, so you can restore the previous behavior easily at a later point.

Once you have done so, set the GRUB_DEFAULT variable to point to the index of the menu and submenu. Note that the index starts at 0, so for a typical setup, the line in /etc/default/grub would look like this:

GRUB_DEFAULT="1>2"

The “1” means the second entry of the main menu (“Advanced options”), the “2” means the third entry of the submenu. Again, update these numbers consistent with your configuration.

Caution: Ensure that you have chosen the right index for the main menu and the submenu, and double-check that you are beginning the count at 0, not 1; otherwise, you may boot into the wrong kernel.

This change still has to be applied to take effect on the next boot:

sudo update-grub

Now you can reboot into the old, working kernel.

sudo reboot

The server should come up automatically. From here on, you should be able to perform all administrative tasks via SSH as usual. If you want additional confirmation of the kernel version, the command `uname -r` should display the expected kernel version number.

Please notify us of the compatibility issue so we can help you resolve it ASAP.

## 41.5 Report Compatibility Issues

If you have encountered issues with a kernel update, it is important that you report them to us so that we may incorporate any necessary changes to our updated kernel, and so that we can work with you to switch back to the new kernel as soon as possible.

Run the following commands via SSH from the Admin Workstation:

```bash
cd ~/Persistent/securedrop/
source .venv/bin/activate
cd install_files/ansible-base
ansible all -b -m setup > server-facts.log
```

Please also send us a copy of `/var/log/syslog` and `/var/log/dmesg` for analysis.

You can share `server-facts.log`, `syslog` and `dmesg` with us as follows:

- If you are a member of our Support Portal, please create a new issue and attach the files to it.
- Alternatively, email us at securedrop@freedom.press (GPG encrypted) with the subject “SecureDrop kernel facts” and the files attached.

Once we get your information, we can try to provide assistance to resolve compatibility issues.
If you are not a member of our Support Portal, we also encourage you to request help in the SecureDrop Community Forums. Choose carefully what information to disclose publicly. For example, raw logs may contain sensitive information useful to potential attackers.

## 41.6 Test and Enable an Updated Kernel

If you have changed your default kernel, we urge you to test an updated kernel as soon as it becomes available in a future SecureDrop release. Note that an update may be enforced as part of a release to protect the security of your instance. Please consult the release notes for details about kernel updates.

You can test a kernel update without downtime for your instance by booting your Monitor Server with the new kernel. Log into your Monitor Server using the Admin Workstation. Shut down the server safely using the command `sudo poweroff`. Ensure that the server is fully powered off.

Attach required peripherals and power the server back up. After the GRUB bootloader appears, select Advanced options for Ubuntu, pictured below.

```
GNU GRUB  version 2.02~beta2-9ubuntu1.14

   Ubuntu
   *Advanced options for Ubuntu
   Memory test (memtest86+)
   Memory test (memtest86+, serial console 115200)
```

Use the ↑ and ↓ keys to select which entry is highlighted. Press enter to boot the selected OS, `e` to edit the commands before booting or `c` for a command-line.

If a SecureDrop release with a kernel update has been installed on your system, the updated kernel version will be available in the list of options:
Select the new kernel (you do not need to use the version with recovery mode). If you do not know your admin account password, you can boot into single user mode by editing the boot options. Otherwise, press enter to boot.

Verify that you can boot successfully, and that you have network access (`sudo host freedom.press`). If you still encounter problems with the new kernel, please report compatibility issues at your earliest convenience, and reboot the server into the old kernel for now.

If the update resolved compatibility issues with an earlier kernel version, you can make the new kernel the default. Edit the file `/etc/default/grub`, e.g., by issuing the following command:

```
sudo nano /etc/default/grub
```

Make a backup of the file or take a note of the current value of `GRUB_DEFAULT` somewhere, so you can restore the previous behavior if needed. Change the line to `GRUB_DEFAULT=0`. This configures the bootloader to default to loading the most recent kernel version installed on your server.

This change still has to be applied to take effect on the next boot:

```
sudo update-grub
```

Safely shut down the Monitor Server, remove attached peripherals, and reboot it. Verify that it is working correctly by logging in using your Admin Workstation. If everything is working as expected, you can make the same change to `/etc/default/grub` on your Application Server as well. Remember to again run the command `sudo update-grub` when you are done.

You can make the change on the Application Server from your Admin Workstation and reboot the server using the command `sudo reboot`.

Subsequent kernel updates will again be applied automatically.
There are a variety of support options available for people who need help installing SecureDrop, or are looking for help with their existing SecureDrop instance.

### 42.1 Community Based Support

The [SecureDrop forum](https://discourse.secure.drop) is a great place to discuss SecureDrop and to get help from others. It is based on Discourse and creating an account is simple and easy.

Additionally, the [SecureDrop Gitter channel](https://gitter.im/SecureDrop) is a great place to discuss SecureDrop in real-time chat. This is mostly a development focused channel, but occasionally support questions do come up.

**Warning:** Remember that both the SecureDrop forum and the Gitter channel are public. Do not post any sensitive information through public channels.

### 42.2 Priority Support and Training

Freedom of the Press Foundation provides paid priority support and SecureDrop training to organizations. Visit the [Priority Support and Training](https://secure.drop.io/priority-support) pages on the SecureDrop website for more information.
SecureDrop v3 Onion Services

Tor onion services provide anonymous inbound connections to websites and other servers exclusively over the Tor network. For example, SecureDrop uses onion services for the Journalist and Source Interface websites, as well as for administrative access to the servers in SSH-over-Tor mode.

Previous to the release of SecureDrop 1.0.0, only v2 onion services were available, but administrators now have the option of enabling next-generation (v3) onion services. V3 onion services use stronger cryptographic primitives than v2 onion services, and include redesigned protocols that guard against service information leaks on the Tor network. The unique identifier in V3 onion addresses is 56 characters long - for example:

```
http://vww6ybal4bd7szmgncyruucpqfkgqahzddi37ktce03ah7ngmcopnpyyd.onion/
```

This makes them easily distinguishable from 16-character v2 onion addresses, such as:

```
http://secrdrop5wyphb5x.onion/
```

For new SecureDrop instances, it is recommended to enable v3 onion services only. For existing instances, migration to v3 onion services is recommended, and the ability to run both v2 and v3 onion services simultaneously is supported to make the process easier.

### 43.1 Migrating from v2 to v3 only or v2+v3

**Important:** In a future release of SecureDrop, v3 onion services will be required rather than optional, and there is no automatic downgrade path from v3 to v2 using the securedrop-admin tool. It is recommended that you follow the v2+v3 migration path and test v3 functionality thoroughly before disabling v2 services.

#### 43.1.1 Preparing for the migration

Before starting the migration process, you should decide whether to move straight to v3 only, or move to v2+v3 temporarily. As the URLs of your onion services will change with the move to v3, moving to v2+v3 first will allow
you to minimize the impact of the migration on sources and journalists.

URL changes will affect the following:

- The Source Interface address will change - once the migration is complete, you will need to update your landing page and other resources that reference the address, such as your SecureDrop directory entry.
- Journalist and Admin Workstations will need to be updated to use the v3 addresses of the Journalist and Source Interface, and the SSH proxy services if your instance is using SSH over Tor.
- If your instance uses HTTPS, you will need to provision a new certificate for the v3 Source Interface address - this will need to be done after the new address has been generated.

Note: If your certificate provisioning process requires validation of the new v3 domain, you may not be able to complete the v3 migration process without first disabling HTTPS for v2. If your instance currently uses HTTPS with an EV certificate, please contact us via the SecureDrop support portal or via email to securedrop@freedom.press before proceeding with the migration. Please use our GPG key for any email communication.

Before proceeding with the migration, you should also back up the instance and Admin Workstation USB - for more information, see the following instructions:

- Back up the instance.
- Back up the Admin Workstation.

### 43.1.2 Enabling v3 onion services

To enable v3 onion services, you will need to run the installation playbook, via the `./securedrop-admin install` command, and then update the Admin Workstation with `./securedrop-admin tailsconfig`.

- First, boot into the Admin Workstation with the persistent volume unlocked and an admin password set.
- Next, open a terminal via Applications > Favorites > Terminal and change the working directory to the SecureDrop application directory:

```bash
cd ~/Persistent/securedrop
```

- Verify that SecureDrop version 1.0.0 or greater is available or installed on your instance with the command:

```bash
cd ~
ssh app apt-cache policy securedrop-app-code
```

Version 1.0.0 should be listed as installed or as an installation candidate.

- Verify that the Admin Workstation’s SecureDrop code is on 1.0.0 or greater, using the GUI updater or the command:

```bash
./securedrop-admin update
```

- After updating the latest SecureDrop version, use the following command to update securedrop-admin’s dependencies:

```bash
./securedrop-admin setup
```

- Next, enable v3 onion services (and optionally disable v2 services) using:

```bash
./securedrop-admin sdconfig
```
This command will step through the current instance configuration. None of the current settings should be changed. When prompted to enable v2 and v3 services, you should choose either yes to both to use v2 and v3 concurrently, or no to v2 and yes to v3 to migrate to v3 only.

- Once the configuration has been updated, run the installation playbook using the command:

```bash
./securedrop-admin install
```

This will enable v3 onion services on the Application and Monitor Servers.

- When the installation playbook run is complete, update the Admin Workstation to use v3 onion services using the command:

```bash
./securedrop-admin tailsconfig
```

- Next, verify connectivity between the Admin Workstation and the SecureDrop instance as follows:
  - Use the Source desktop shortcut to connect to the Source Interface and verify that the new 56-character address is present in the Tor Browser address bar.
  - Use the Journalist desktop shortcut to connect to the Journalist Interface and verify that the new 56-character address is present in the Tor Browser address bar.
  - Use the commands `ssh app` and `ssh mon` in a terminal to verify SSH access to the Application and Monitor Servers.

- Finally, back up the instance and Admin Workstation USB.

### 43.1.3 (Optional) enabling HTTPS

If your instance serves the Source Interface over HTTPS, and you plan to continue using HTTPS with v3 onion services, you’ll need to provision a new certificate for the new v3 address.

You’ll find the new Source Interface address in the file:

```bash
~/.persistent/securedrop/install_files/ansible-base/app-sourcev3-ths
```

Follow the instructions in HTTPS on the Source Interface to provision and install the new certificate.

### 43.1.4 Updating Workstation USBs

If you chose to keep v2 enabled, Admin and Journalist Workstations that have not yet been updated will still be able to connect to the v2 onion services. Even so, you should update all workstations to use v3 services as soon as possible.

**Journalist Workstation:**

- In the Admin Workstation used to enable v3 onion services, copy the following files to an encrypted Transfer Device:

```bash
~/.persistent/securedrop/install_files/ansible-base/app-sourcev3-ths
~/.persistent/securedrop/install_files/ansible-base/app-journalist.auth_private
```

- Then, boot into the Journalist Workstation to be updated, with the persistent volume unlocked and an admin password set.

- Next, open a terminal via Applications > Favorites > Terminal and change the working directory to the SecureDrop application directory:
cd ~/Persistent/securedrop

- Ensure that the SecureDrop application code has been updated to the latest version, using either the GUI updater or the ./securedrop-admin update command.
- Insert the Transfer Device. Copy the app-sourcev3-ths and app-journalist.auth_private files from the Transfer Device to ~/ Persistent/securedrop/install_files/ansible-base.
- Open a terminal and run ./securedrop-admin tailsconfig to update the SecureDrop desktop shortcuts.
- Verify that the new 56-character addresses are in use by visiting the Source and Journalist Interfaces via the SecureDrop desktop shortcuts.
- Securely wipe the files on the Transfer Device, by right-clicking them in the file manager and selecting Wipe.

Admin Workstation:

- In the Admin Workstation used to enable v3 onion services, copy the following files to an encrypted Transfer Device:

```
~/Persistent/securedrop/install_files/ansible-base/app-sourcev3-ths
~/Persistent/securedrop/install_files/ansible-base/app-journalist.auth_private
~/Persistent/securedrop/install_files/ansible-base/app-ssh.auth_private
~/Persistent/securedrop/install_files/ansible-base/mon-ssh.auth_private
~/Persistent/securedrop/install_files/ansible-base/tor_v3_keys.json
~/Persistent/securedrop/install_files/ansible-base/group_vars/all/site-specific
```

If your instance uses SSH over Tor, also copy the following files:

```
~/Persistent/securedrop/install_files/ansible-base/app-sourcev3-ths
~/Persistent/securedrop/install_files/ansible-base/app-ssh.auth_private
```

- Then, boot into the Admin Workstation to be updated, with the persistent volume unlocked and an admin password set.
- Next, open a terminal via Applications > Favorites > Terminal and change the working directory to the SecureDrop application directory:

```
cd ~/Persistent/securedrop
```

- Ensure that the SecureDrop application code has been updated to the latest version, using either the GUI updater or the ./securedrop-admin update command.
- Insert the Transfer Device. Copy the app-sourcev3-ths, *.auth_private, and tor_v3_keys.json files from the Transfer Device to ~/Persistent/securedrop/install_files/ansible-base.
- Copy the site-specific file from the Transfer Device to ~/Persistent/securedrop/install_files/ansible-base/group_vars/all.
- Open a terminal and run ./securedrop-admin tailsconfig to update the SecureDrop desktop shortcuts.
- Verify that the new 56-character addresses are in use by visiting the Source and Journalist Interfaces via the SecureDrop desktop shortcuts.
- Verify that ~/.ssh/config contains the new 56-character addresses for the app and mon host entries, and that the Application and Monitor Servers are accessible via ssh app and ssh mon respectively.
- Securely wipe the files on the Transfer Device, by right-clicking them in the file manager and selecting Wipe.
43.1.5 Updating Source Interface references

In order for sources to find and use the new v3 Source Interface, you’ll need to update your landing page. If your instance details are listed anywhere else (for example, in the SecureDrop directory), you should update those listings too.

You’ll find the new Source Interface address in the file:

~/Persistent/securedrop/install_files/ansible-base/app-sourcev3-ths

43.1.6 Disabling v2 onion services

Once you’ve successfully enabled v3 onion services, and updated your workstations, you should disable v2 onion services altogether.

First, it’s recommended that you coordinate with the journalists using the instance to ensure that any ongoing source conversations are uninterrupted. They can use SecureDrop’s reply feature to give active sources advance notice of the address change.

When you’re ready, follow the steps below to transition to v3 services only:

- First, boot into the Admin Workstation with the persistent volume unlocked and an admin password set.
- Open a terminal and change the working directory to the SecureDrop application directory with the command:

  ```
  cd ~/Persistent/securedrop
  ```

- Next, update the application configuration using the command:

  ```
  ./securedrop-admin sdconfig
  ```

  This command will step through the current instance configuration. When prompted you should type no for v2 services and yes for v3 services to migrate to v3 only. No other settings should be modified.

- Once the configuration has been updated, run the installation playbook using the command:

  ```
  ./securedrop-admin install
  ```

  This will disable v2 onion services on the Application and Monitor Servers.

- When the installation playbook run is complete, update the Admin Workstation to use v3 onion services only using the command:

  ```
  ./securedrop-admin tailsconfig
  ```

- Next, verify connectivity between the Admin Workstation and the SecureDrop instance, checking the desktop shortcuts and SSH access.

- Then back up the instance and Admin Workstation USB.

- Finally, update your other Admin Workstations: from a terminal, run:

  ```
  ./securedrop-admin sdconfig  # choose "no" for v2, "yes" for v3
  ./securedrop-admin tailsconfig
  ```

43.1. Migrating from v2 to v3 only or v2+v3
44.1 Updating the Tails Workstations

We recommend that you update all Tails drives to version 3.16, which was released on September 4, 2019. Follow the Tails graphical prompts on your workstations to perform this upgrade.

On a subsequent boot of your SecureDrop Journalist and Admin Workstations, the SecureDrop Workstation Updater will alert you to workstation updates. You must have configured an administrator password on the Tails welcome screen in order to use the graphical updater.

**Note:** If you have not previously updated this workstation to SecureDrop 0.14.0, we recommend following the manual update procedure. The manual procedure enforces the use of a secure keyserver.

If you are unsure which version of SecureDrop a workstation is using, you can run the command `git status` in `~/Persistent/securedrop`. The first line of the output will tell you the current version, e.g., “HEAD detached at 0.14.0”.

If the workstation is on SecureDrop 0.14.0, it is safe to use the graphical updater by clicking “Update Now”: 
Please note that this only updates the SecureDrop code on your Tails workstations. Tails updates must be performed separately.
44.1.1 Manually updating the SecureDrop code

If you are running an older version of SecureDrop on this workstation, or if the automated update fails, you can perform a manual update by running the following commands:

```
cd ~/Persistent/securedrop
git fetch --tags
gpg --keyserver hkps://keys.openpgp.org --recv-key "2224 5C81 E3BA EB41 38B3 6061 310F 5612 00F4 AD77"
git tag -v 1.0.0
```

The output should include the following two lines:

```
gpg: using RSA key 22245C81E3BAEB4138B36061310F561200F4AD77
gpg: Good signature from "SecureDrop Release Signing Key"
```

Please verify that each character of the fingerprint above matches what on the screen of your workstation. If it does, you can check out the new release:

```
git checkout 1.0.0
```

**Important:** If you do see the warning “refname ‘1.0.0’ is ambiguous” in the output, we recommend that you contact us immediately at securedrop@freedom.press (GPG encrypted).

Finally, run the following commands:

```
./securedrop-admin setup
./securedrop-admin tailsconfig
```

44.2 Backing Up the Tails Workstations

USB flash drives degrade over time and vary in quality. To ensure continued access to SecureDrop by administrators and journalists, we recommend backing up the Tails Workstations on the occasion of a new SecureDrop release, after you have completed the upgrade process for each drive.

You can use a single storage device for backups of multiple workstations. See our [Workstation Backup Guide](#) for more information.

44.3 Enabling v3 Onion Services

SecureDrop 1.0.0 is the first release to support v3 onion services. Version 3 of the protocol for onion services offers significant security benefits, and we recommend that you transition your SecureDrop installation to v3 in the near future.

Enabling v3 onion services will create new, 56 character .onion addresses for your *Source Interface*, your *Journalist Interface*, and for SSH access (if SSH access over Tor is enabled). This means that you will have to update the .onion address advertised on your SecureDrop landing page after switching to v3.

For testing purposes, you can enable v3 alongside v2 before switching fully to v3. Please see our [documentation for v3 onion services](#) for more information.
44.4 Logo Update

SecureDrop ships with a new default logo for the SecureDrop project. This will not affect your installation if you have already customized the logo of your instance. If you have not done so yet, we recommend uploading a custom logo by following the instructions.

If you currently show the SecureDrop logo on your landing page or in other materials related to SecureDrop, please update it to the new version. See our brand refresh guide (PDF) for more information, and please do not hesitate to reach out if we can help in any way.

44.5 Getting Support

Should you require further support with your SecureDrop installation, we are happy to help!

- Community support is available at https://forum.securedrop.org
- If you are already a member of our support portal, please don’t hesitate to open a ticket there. If you would like to request access, please contact us at securedrop@freedom.press (GPG encrypted).
- The Freedom of the Press Foundation offers training and priority support services. See https://securedrop.org/priority-support/ for more information.
CHAPTER 45

Upgrade from 0.13.1 to 0.14.0

45.1 Updating the Tails Workstations

We recommend that you update all Tails drives to version 3.15, which was released on July 9, 2019. Follow the Tails graphical prompts on your workstations to perform this upgrade.

On a subsequent boot of your SecureDrop Journalist and Admin Workstations, the SecureDrop Workstation Updater will alert you to workstation updates. Due to a recent surge in attacks against GPG keyservers, we recommend against using the graphical updater for this release.

We have uploaded the SecureDrop Release Key to the new keys.openpgp.org keyserver, which includes important mitigations against this type of attack. In the process, we have also updated the expiration date of the release key (the fingerprint remains the same).

To update your workstations using the new keyserver, run the following commands:

```
cd ~/Persistent/securedrop
git fetch --tags
gpg --keyserver hkps://keys.openpgp.org --recv-key "2224 5C81 E3BA EB41 38B3 6061 310F 5612 00F4 AD77"
git tag -v 0.14.0
```

The output should include the following two lines:

```
gpg: using RSA key 22245C81E3BAEB4138B36061310F561200F4AD77
gpg: Good signature from "SecureDrop Release Signing Key"
```

Please verify that each character of the fingerprint above matches what on the screen of your workstation. If it does, you can check out the new release:

```
git checkout 0.14.0
```

**Important:** If you do see the warning “refname ‘0.14.0’ is ambiguous” in the output, we recommend that you contact
us immediately at securedrop@freedom.press (GPG encrypted).

Finally, run the following commands:

```
./securedrop-admin setup
./securedrop-admin tailsconfig
```

Once you have updated the workstation, the graphical updater will always use the new keys.openpgp.org keyserver.

### 45.2 Backing Up the Tails Workstations

USB flash drives degrade over time and vary in quality. To ensure continued access to SecureDrop by administrators and journalists, we recommend backing up the Tails Workstations on the occasion of a new SecureDrop release, after you have completed the upgrade process for each drive.

You can use a single storage device for backups of multiple workstations. See our [Workstation Backup Guide](#) for more information.

### 45.3 Getting Support

Should you require further support with your SecureDrop installation, we are happy to help!

- Community support is available at [https://forum.securedrop.org](https://forum.securedrop.org)

- If you are already a member of our support portal, please don’t hesitate to open a ticket there. If you would like to request access, please contact us at securedrop@freedom.press (GPG encrypted).

46.1 Updating the Tails Workstations

If you have not already done so, we recommend that you update all Tails drives to version 3.14, which was released on May 21, 2019. Follow the Tails graphical prompts on your workstations to perform this upgrade.

On a subsequent boot of your SecureDrop Journalist and Admin Workstations, the SecureDrop Workstation Updater will alert you to workstation updates. You must have configured an administrator password on the Tails welcome screen in order to use the graphical updater.

Choose “Update Now” on each of the workstations:
Please note that this only updates the SecureDrop code on your Tails workstations. Tails upgrades must be performed separately.
If you don’t see the graphical updater, or if the automated update fails, you can perform a manual update by running the following commands:

```
cd ~/Persistent/securedrop
git fetch --tags
gpg --recv-key "2224 5C81 E3BA EB41 38B3 6061 310F 5612 00F4 AD77"
git tag -v 0.13.1
```

**Note:** You may have to run the `--recv-key` command repeatedly for it to work.

The output should include the following two lines:

```
gpg: using RSA key 22245C81E3BAEB4138B36061310F561200F4AD77
gpg: Good signature from "SecureDrop Release Signing Key"
```

Please verify that each character of the fingerprint above matches what on the screen of your workstation. If it does, you can check out the new release:

```
git checkout 0.13.1
```

**Important:** If you do see the warning “refname ‘0.13.1’ is ambiguous” in the output, we recommend that you contact us immediately at securedrop@freedom.press (GPG encrypted).

Finally, run the following commands:

```
./securedrop-admin setup
./securedrop-admin tailsconfig
```

## 46.2 Getting Support

Should you require further support with your SecureDrop installation, we are happy to help!

- Community support is available at [https://forum.securedrop.org](https://forum.securedrop.org)
- If you are already a member of our support portal, please don’t hesitate to open a ticket there. If you would like to request access, please contact us at securedrop@freedom.press (GPG encrypted).
As of **May 1, 2019**, Ubuntu 14.04 has reached End of Life. If you are still running Ubuntu 14.04 on your Application and Monitor Server, your servers will no longer receive security updates for operating system packages, the kernel, or SecureDrop itself.

That means that a sufficiently severe vulnerability discovered in any of those components may permit an adversary to compromise SecureDrop servers running Ubuntu 14.04.

For this reason, starting May 1, we recommend a reinstall on Ubuntu 16.04 for any SecureDrop still on Ubuntu 14.04. See our installation guide.

This will result in a new .onion address for your Source Interface and your Journalist Interface. You will need to create new user accounts and USB drives for administrators and journalists, and sources you are currently in touch with will no longer be able to log in using their codename. We recommend notifying your sources about this change on your Landing Page.

Unless you have reason to believe that the Submission Key may have been compromised, you do **not** need to reinstall the Secure Viewing Station. Instead, during this part of the installation process, use a copy of your public key obtained from your Secure Viewing Station.

### 47.1 Saving old submissions

If you require access to old submissions to your SecureDrop, you need to save them securely. We do not recommend using the standard backup procedure via securedrop-admin backup, as restoring such a backup will reinstate secrets and credentials that may have been compromised.

Instead, download any submissions you have not already downloaded to your Secure Viewing Station following the standard process as described in the Journalist Guide. If you do not reinstall the Secure Viewing Station, you will be able to continue to view these submissions on the Secure Viewing Station as before, but you will no longer be able to reply to the sources that sent them, until they create a new account.
Caution: If you do reinstall your Secure Viewing Station, you **must** copy the public and private Submission Key from the old Secure Viewing Station to the new one. Without the keypair, you will not be able to decrypt old submissions on a new Secure Viewing Station.

47.2 Contact us

If you have questions or comments regarding this process, please don’t hesitate to reach out:

- Community support is available at [https://forum.securedrop.org](https://forum.securedrop.org)
- If you are already a member of our support portal, please don’t hesitate to open a ticket there. If you would like to request access, please contact us at securedrop@freedom.press (GPG encrypted).
Contributing to SecureDrop

Thank you for your interest in contributing to SecureDrop! We welcome both new and experienced open-source contributors and are committed to making it as easy as possible to contribute. Whether you have a few minutes or many hours, there are a variety of ways to help. We are always looking for help from:

- programmers, to help us develop SecureDrop;
- release managers, to create and maintain Debian GNU/Linux packages and repositories;
- writers, to help improve the documentation;
- translators, to translate SecureDrop;
- designers, to help improve the source and journalist web interfaces and Tails customization;
- moderators and support volunteers, to help with the support forums.

You can always find a developer to answer any questions you may have on the SecureDrop instant messaging channel. You can also register on the forum for more information and to participate in longer discussions.

**Note:** Not sure where to start? You can always ask for advice in the chat room.

### 48.1 Programmers

The SecureDrop system includes Flask-based web applications for sources and journalists. It is deployed across multiple machines with Ansible. Most of SecureDrop’s code is written in Python.

The following links should help you find something to work on:

#### 48.1.1 Bugs

- High-priority bugs
- Middle-priority bugs
• Low-priority bugs

48.1.2 Issues Sorted by Topic

• User experience
• Internationalization (i18n)
• Source and journalist applications
• Application code cleanup
• SecureDrop Workstation
• Source experience
• Journalist experience
• Ansible logic/installation
• Operations and deployment
• Threat model
• IDS noise
• OSSEC
• Security
• Research
• Developer workflow
• Tests
• Continuous Integration

When you’re ready to share your work with the SecureDrop team for review, submit a pull request with the proposed changes. Tests will run automatically on GitHub.

If you would like to contribute on a regular basis, you’ll want to read the developer documentation and set up a local development environment to preview changes, run tests locally, etc.

48.2 Writers

Technical writers and editors are invited to review the documentation and fix any mistakes in accordance with the documentation guidelines.

If this is your first time helping with SecureDrop documentation, consider working on low-hanging fruit to become familiar with the process.

48.2.1 Documentation Issues

• High-priority
• Middle-priority
• Low-priority
48.3 DevOps

The SecureDrop web site and the GitHub repository are controlled and maintained by Freedom of the Press Foundation employees.

48.4 Release Managers

All software deployed with SecureDrop is installed via Debian GNU/Linux packages via Ansible. The primary repository is controlled, maintained, and signed by Freedom of the Press Foundation employees. The current responsibilities of the release manager are covered in detailed documentation.

If you are a Debian developer you can help improve packaging and the release process:

- Building SecureDrop application and OSSEC packages and pending bugs and tasks
- Building grrsecurity kernels and pending bugs and tasks

48.5 Translators

Translating SecureDrop is crucial to making it useful for investigative journalism around the world. If you know English and another language, we would welcome your help.

SecureDrop is translated using Weblate. We provide a detailed guide for translators, and feel free to contact us in the translation section of the SecureDrop forum for help. Non-English forum discussions are also welcome.

48.6 Designers

If you have web design or user experience design skills, take a look at these issues on GitHub:

- User experience
- CSS/SASS and HTML
- Journalist user experience

48.7 Moderators and Support

Those running a production instance of SecureDrop are encouraged to read the support documentation to get help from the Freedom of the Press Foundation. For less sensitive topics such as running a demo or getting help to understand a concept, a public forum section is better suited. To assist on the forum:

- Look for the latest unanswered questions in the forum and answer them.
• If you find questions elsewhere in the forum that have a better chance at getting an answer in the support section, suggest in Gitter to move topics from a category to another.
Setting Up the Development Environment

**Note:** SecureDrop maintains two versions of documentation: **stable** and **latest**. **stable** is the default used by our Read the Docs site, and is built from our latest signed git tag. **latest** is built from the head of the **develop** git branch. In almost all cases involving development work, you'll want to make sure you have the **latest** version selected by using the menu in the bottom left corner of the Read the Docs site.

### 49.1 Prerequisites

SecureDrop is a multi-machine design. To make development and testing easy, we provide a set of virtual environments, each tailored for a specific type of development task. We use Vagrant, VirtualBox, and Docker and our Ansible playbooks can provision these environments on either virtual machines or physical hardware.

**Note:** SecureDrop is written in Python 2. We plan to migrate to Python 3 in a future release. Please use a Python 3 virtualenv for development, while ensuring that your changes are compatible with both Python 3 and Python 2, to ensure they work in production and CI, both of which use Python 2.7.

### 49.2 Quick Start

The Docker based environment is suitable for developing the web application and updating the documentation.

#### 49.2.1 Ubuntu or Debian GNU/Linux

```
sudo apt-get update
sudo apt-get install -y make git
```
We recommend using the stable version of Docker CE (Community Edition) which can be installed via the official documentation links:

- Docker CE for Ubuntu
- Docker CE for Debian

Make sure to follow the Post-installation steps for Linux, as well.

### 49.2.2 macOS

Install Docker.

### 49.2.3 Qubes

Create a StandaloneVM based on Debian 9, called `sd-dev`. You can use the Q menu to configure a new VM, or run the following in `dom0`:

```bash
qvm-clone --class StandaloneVM debian-9 sd-dev
qvm-start sd-dev
qvm-sync-appmenus sd-dev
```

The commands above will create a new StandaloneVM, boot it, then update the Qubes menus with applications within that VM. Open a terminal in `sd-dev`, and proceed with installing Docker CE for Debian.

### 49.2.4 Fork & Clone the Repository

Now you are ready to get your own copy of the source code. Visit our repository fork it and clone it on your local machine:

```bash
git clone git@github.com:<your_github_username>/securedrop.git
```

### 49.2.5 Using the Docker Environment

The Docker based helpers are intended for rapid development on the SecureDrop web application and documentation. They use Docker images that contain all the dependencies required to run the tests, a demo server etc.

**Tip:** When run for the first time, building Docker images will take a few minutes, even one hour when your Internet connection is not fast. If you are unsure about what happens, you can get a more verbose output by setting the environment variable `export DOCKER_BUILD_VERBOSE=true`.

The SecureDrop repository is bind mounted into the container and files modified in the container are also modified in the repository. This container has no security hardening or monitoring.

To get started, you can try the following:

```bash
cd securedrop/securedrop
make dev # run development servers
make test # run tests
bin/dev-shell bin/run-test tests/functional # functional tests only
bin/dev-shell bash # shell inside the container
```
To specify the version of Python you want to use, set the PYTHON_VERSION environmental variable, like so:

```
PYTHON_VERSION=3 make test  # Run tests on Python 3
PYTHON_VERSION=2 make test  # Run tests on Python 2
PYTHON_VERSION=3 make dev  # Run dev container on Python 3
PYTHON_VERSION=2 make dev  # Run dev container on Python 2
```

**Tip:** The interactive shell in the container does not run redis, Xvfb etc. However you can import shell helper functions with `source bin/dev-deps` and call `run_xvfb`, `maybe_create_config_py` etc.

SecureDrop consists of two separate web applications (the Source Interface and the Journalist Interface) that run concurrently. In the development environment they are configured to detect code changes and automatically reload whenever a file is saved. They are made available on your host machine by forwarding the following ports:

- **Source Interface:** localhost:8080
- **Journalist Interface:** localhost:8081

You should use Tor Browser to test web application changes, see here for instructions.

A test administrator (journalist) and non-admin user (dellsberg) are created by default when running `make dev`. In addition, sources and submissions are present. The test users have the following credentials. Note that the password and TOTP secret are the same for both accounts for convenience during development.

- **Username:** journalist or dellsberg
- **Password:** correct horse battery staple profanity oil chewy
- **TOTP secret:** JHCO GO7V CER3 EJ4L

If you need to generate the six digit token, use the TOTP secret in combination with an authenticator application that implements RFC 6238, such as FreeOTP (Android and iOS) or oathtool (command line tool, multiple platforms). Instead of typing the TOTP code, you can simply scan the following QR code:
You can also generate the 2FA code using the Python interpreter:

```python
>>> import pyotp
>>> pyotp.TOTP('JHCOGO7VCER3EJ4L').now()
'u'422038'
```

### 49.3 Setting Up a Multi-Machine Environment

**Note:** You do not need this step if you only plan to work on the web application or the documentation.

To get started, you will need to install Vagrant, VirtualBox, Docker, and Ansible on your development workstation.

#### 49.3.1 Ubuntu or Debian GNU/Linux

**Note:** Tested on: Ubuntu 16.04 and Debian GNU/Linux stretch
sudo apt-get update
sudo apt-get install -y build-essential libssl-dev libffi-dev python-dev \
        dpkg-dev git linux-headers-$\{uname -r\} virtualbox

We recommend using the latest stable version of Vagrant, 1.8.5 at the time of this writing, which might be newer than what is in your distro’s package repositories. Older versions of Vagrant has been known to cause problems (GitHub #932, GitHub #1381). If `apt-cache policy vagrant` says your candidate version is not at least 1.8.5, you should download the current version from the Vagrant Downloads page and then install it.

```bash
# If your OS vagrant is recent enough
sudo apt-get install vagrant
# OR this, if you downloaded the deb package.
sudo dpkg -i vagrant.deb
```

**Warning:** We do not recommend installing vagrant-cachier. It destroys apt’s state unless the VMs are always shut down/rebooted with Vagrant, which conflicts with the tasks in the Ansible playbooks. The instructions in Vagrantfile that would enable vagrant-cachier are currently commented out.

VirtualBox should be at least version 5.x. See GitHub #1381 for documentation of incompatibility with the older VirtualBox 4.x release series.

Finally, install Ansible so it can be used with Vagrant to automatically provision VMs. We recommend installing Ansible from PyPi with `pip` to ensure you have the latest stable version.

```bash
sudo apt-get install python-pip
```

The version of Ansible recommended to provision SecureDrop VMs may not be the same as the version in your distro’s repos, or may at some point flux out of sync. For this reason, and also just as a good general development practice, we recommend using a Python virtual environment to install Ansible and other development-related tooling. Using `virtualenvwrapper`:

```bash
sudo apt-get install virtualenvwrapper
source /usr/share/virtualenvwrapper/virtualenvwrapper.sh
mkvirtualenv -p /usr/bin/python3 securedrop
```

**Note:** You’ll want to add the command to source `virtualenvwrapper.sh` to your `~/.bashrc` (or whatever your default shell configuration file is) so that the command-line utilities `virtualenvwrapper` provides are automatically available in the future.

### 49.3.2 macOS

Install the dependencies for the development environment:

1. Vagrant
2. VirtualBox
3. Ansible
4. rsync >= 3.1.0

If you use homebrew-cask to manage macOS apps, you can install Vagrant and VirtualBox that way. As for Ansible, we strongly recommend installing it in a virtual environment using `virtualenvwrapper` and `pip`, so as not to
install the older version we use system-wide. The following commands assume your default Python is the Python 2 that ships with macOS. If you are using a different version, the path to virtualenvwrapper.sh will differ. Running `pip show virtualenvwrapper` should help you find it.

```
sudo easy_install pip  # if you don't already have pip
sudo -H pip install -U virtualenvwrapper --ignore-installed six
source /usr/local/bin/virtualenvwrapper.sh
mkvirtualenv -p python securedrop
```

**Note:** You’ll want to add the command to source `virtualenvwrapper.sh` to your `~/.bashrc` (or whatever your default shell configuration file is) so that the command-line utilities `virtualenvwrapper` provides are automatically available in the future.

The version of rsync installed by default on macOS is extremely out-of-date, as is Apple’s custom. We recommend using Homebrew to install a modern version (3.1.0 or greater): `brew install rsync`.

### 49.3.3 Fork & Clone the Repository

Now you are ready to get your own copy of the source code. Visit our repository fork it and clone it on you local machine:

```
git clone git@github.com:<your_github_username>/securedrop.git
```

### 49.3.4 Install Python Requirements

SecureDrop uses many third-party open source packages from the python community. Ensure your virtualenv is activated and install the packages.

```
pip install -r securedrop/requirements/python3/develop-requirements.txt
```

**Note:** You will need to run this everytime new packages are added.
50.1 Forking and Cloning the Project

1. Fork SecureDrop on GitHub from the Main Repository to your own profile.
2. Clone the forked repository.

```
git clone https://github.com/<your-username>/securedrop.git
cd securedrop
```

3. Add the Main Repository as an upstream remote.

```
git remote add upstream https://github.com/freedomofpress/securedrop.git
```

50.2 Make Your Changes and Push to the Fork

50.2.1 Create a Branch

Create a branch on which you make your changes.

```
git checkout -B change-one
```

50.2.2 Make Your Changes and Commit

Now enter the directory of your fork and make changes as you wish. Run tests for the changes you have made.

If you create a new file, remember to add it with `git add`.

```
git add <new-file>
```
Commit your changes, adding a description of what was added. If you’re not used to Git, the simplest way is to commit all modified files and add a description message of your changes in a single command like this:

```
git commit -a -m "<Description of changes made>"
```

### 50.2.3 Pull the Upstream Changes

We get any updates made in the upstream repository.

```
git pull upstream develop
```

### 50.2.4 Rebasing

Rebasing is the process of moving or combining a sequence of commits to a new base commit. Rebasing is most useful and easily visualized in the context of a feature branching workflow.

Assume the following history exists:

```
A---B---C change-one
/ 
D---E---F---G develop
```

From this point, the result of either of the following commands:

```
git rebase develop
```

```
git rebase develop change-one
```

would be:

```
A'--B'--C' change-one
/ 
D---E---F---G develop
```

**Note:** A and A represents the same set of changes, but have different committer information.

### 50.2.5 Pushing the Changes to GitHub Fork

Once your changes are committed and rebased, push the changes to your GitHub fork.

```
git push origin <branch-name>
```

### 50.3 Making a Pull Request to Get Your Changes Merged in develop Branch

1. Through GitHub make a pull request from the branch that you committed your code to.
2. Once PR is made, the Circle CI build server checks all tests and Codecov runs a report on test coverage. The reports are available in the PR page and also emailed to admins.
3. From there, a maintainer will accept your PR or they may request comments for you to address prior to merge. The maintainer may also ask you to squash your commits prior to merge.
The admin directory contains the source of the securedrop-admin script which is used in Tails to perform various administrative tasks. It is a standalone python module which can be tested on Debian GNU/Linux stretch with:

```bash
python bootstrap.py
source .venv/bin/activate
pip install -r requirements-dev.txt
tox
```

A Docker helper is provided to simplify the installation and make it portable on various operating systems.

Run only flake8 with:

```bash
bin/dev-shell tox -e flake8
```

Run only one test foobar with:

```bash
bin/dev-shell tox -e py2 -- -k foobar
```

Docker has the admin directory mounted from the host into the container, at the same location to avoid any trouble with hardcoded absolute paths. It runs with the id of the host user so files created in the container are owned by the host user instead of root. If a script needs root access, it has passwordless sudo permissions.

Convenience Makefile targets are also provided for the most common tasks:

```bash
$ make
Makefile for developing and testing securedrop-admin.
Subcommands:

<table>
<thead>
<tr>
<th>Target</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>help</td>
<td>Print this message and exit.</td>
</tr>
<tr>
<td>test</td>
<td>Run tox</td>
</tr>
<tr>
<td>update-pip-requirements</td>
<td>Updates all Python requirements files via pip-compile.</td>
</tr>
</tbody>
</table>
```
The SecureDropUpdater is a tool used by the journalists and admins, this tool helps them to update their SecureDrop git repository to the latest released tag. It is a GUI tool and it is written using PyQt5 bindings of the Qt framework. This tool is written using Python3.

52.1 Installing the Dependencies in a Virtual Environment

You can use the pipenv tool to create a virtualenv and install the dependencies. The first step is to install pipenv itself. After that, the following commands can be used to create the environment:

```bash
$ pipenv install
$ pipenv shell
```

The first command will create the virtualenv and install the dependencies. The second command is used to enable the Pipenv shell.

**Note:** The Updater GUI does not use a virtual environment on the Tails Workstations. As such, you can only use dependencies present in Tails.

You can run the GUI via:

```bash
$ python3 SecureDropUpdater
```

Note that since the application expects to run in Tails, you should test its functionality in a Tails VM. You can follow the instructions in the Virtualizing Tails guide to set up your Tails VM.

52.2 To Update the UI Design

The design of the GUI is saved in the journalist_gui/mainwindow.ui file. To update the UI, one has to first install qtcreator tool in the system. We are currently using 5.10.1 version of Qt for this project.
If we make any changes to the UI, we will have to use pyuic5 command to update the corresponding Python code.

```bash
$ pyuic5 journalist_gui/mainwindow.ui -o journalist_gui/updaterUI.py
```

### 52.3 Using Resources in the UI

All icons and images for the UI is stored in the `journalist_gui/static` directory. These are known as resources for the project. The `journalist_gui/resources.qrc` file contains the list of current resources for the project. Each resource needs to be defined inside of a `<file></file>`.

Example qrc file:

```xml
<RCC>
    <qresource prefix="/images">
        <file>static/securedrop.png</file>
        <file>static/securedrop_icon.png</file>
    </qresource>
</RCC>
```

We will have to update the corresponding Python file for any change in this resource file. We can do that using the following command:

```bash
$ pyrcc5 journalist_gui/resources.qrc -o journalist_gui/resources_rc.py
```

**Note:** The `updaterUI.py` and `resources_rc.py` files are generated by the tools. So, do not make any changes to these files. Any changes made to these files will be overridden.

**Warning:** As a reviewer of a PR involving changes to this resource file, you should verify the changes to the file by running `pyrcc5` locally.

### 52.4 Adding and Running Test Cases

We have Python unit tests in the `test_gui.py` file. Any change in the actual application code will also require adding new test cases or updating the old ones. You can run the tests using the following command:

```bash
$ python3 test_gui.py
```
Developing the SecureDrop Client Application

As part of the ongoing work to make an integrated journalist-friendly workstation for SecureDrop we have created a native client application to be run within the Qubes operating system. It helps journalists with the most common activities associated with using SecureDrop in a user friendly manner.

Currently the client is alpha quality although work is ongoing in terms of improving features and the user interface. The source code, and related issues are hosted on GitHub.

## 53.1 Developer Setup

Clone the repository:

```
git clone git@github.com:freedomofpress/securedrop-client.git
```

Ensure you have `pipenv` installed:

```
pip install --user pipenv
```

Next, follow the steps in the README of the project for setting up the development environment.

Check you have everything working by running the complete test suite:

```
make check
```

If you see a green report of 100% unit test coverage, you’re good to go!

## 53.2 How to Find Help

If you would like to report a problem submit a new issue.

If you’d like to chat with other developers working on the client drop into our Gitter chat channel for the project.
Every non-public holiday weekday (except Fridays) at 10am (Pacific Time) we take part in a public daily stand-up, usually via a meeting on Jitsi (although the details of each daily meeting are published on the Gitter channel five minutes before the start of the meeting). All are welcome to contribute.

Otherwise, read on.

## 53.3 Client Architecture

The SecureDrop client is a PyQt application. It’s written using Python 3.5 and the Python bindings for the Qt UI framework (PyQt).

In the root directory of the repository are two important directories: `securedrop_client` (containing the application code) and `tests` containing our unit tests. You’ll also find a Makefile in the root directory which defines commands to run commonly needed activities. Type, `make` to find out what commands are available.

The code in the `securedrop_client` namespace is organised in the following way:

- `app.py` - starts and configures the application.
- `logic.py` - contains the application logic, encapsulated in the `Client` class.
- `db.py` - holds all the SQLAlchemy ORM model definitions for interacting with the local Sqlite database.
- `storage.py` - contains the functions needed for interacting with a remote SecureDrop API and the local database.
- `utils.py` - generic utility functions needed throughout the application.
- `gui` - this namespace contains two modules: `main.py` (containing the `Window` class through which all interactions with the user interface should happen) and `widgets.py` (containing all the custom widgets used by the `Window` class to draw the user interface).

We try very hard to keep the application logic and UI code cleanly separated. Furthermore, we try equally hard to ensure the main GUI code always remains unblocked. For instance look at how the `APICallRunner` is used in `logic.py` to make unblocked network calls to the remote API.

We encourage developers to make sure all classes, methods and functions have docstrings describing the intention behind the code. Obviously, it’s important that such docstrings remain up to date as the code evolves.

If possible, please use Python type hints for new code. We’re going to transition the code base to this style in the not-too-distant future.

## 53.4 Tests

The files and directory structure found within the `tests` directory mirrors that of the files and directories in `securedrop_client`. For instance, all the unit tests for the `securedrop_client/logic.py` module can be found in the `tests/test_logic.py` file.

To run the complete test suite simply type:

```
make check
```

Our code style checkers, full test suite and coverage checker will run and report any errors.

We use the PyTest testing framework for writing and running our unit tests. We expect every test to have an associated comment which describes the intent of the test. As far as possible, tests should be self contained with all the context needed to understand them within each individual unit test (this makes it easier to debug things when the test suite fails as the codebase evolves).
Take a look in any of the test files to see the sort of code we expect for unit tests.
We currently have, and expect to maintain, 100% unit test coverage of our code base. If you’re unsure how to achieve this, please don’t hesitate to get in touch via Gitter or mention this in your description of any pull requests you submit.

## 53.5 Contributing

Our open issues are on GitHub.

Please remember that we have a code of conduct and expect all contributors to abide by it.

Before submitting a pull request, make sure the test suite passes (make check), because our CI tools will flag broken tests before we’re able to merge your code into master.

Most of all, please don’t hesitate to get in touch if you need help, advice or would like guidance.

Thank you for your support!
This document describes the endpoints for SecureDrop’s Journalist Interface API.

54.1 Versioning

The API is versioned and we are currently using version 1. This is set via the base URL, which is:

/api/v1/

54.2 Content Type

Clients shall send the following headers:

'Accept': 'application/json',
'Content-Type': 'application/json'

54.3 Authentication

POST /api/v1/token to get a token with the username, password, and 2FA token in the request body:

```
{
    "username": "journalist",
    "passphrase": "monkey potato pizza quality silica growing deduce",
    "one_time_code": "123456"
}
```

This will produce a response with your Authorization token:
Thereafter in order to authenticate to protected endpoints, send the token in HTTP Authorization header:

```
Authorization: Token eyJhbGciOiJIUzI1NiIsImV4cCI6MTUzMDU4NjU4MiwifWF0IjoxNTMwNTc5MzgyfQ.eyJpZCI6MX0.
```

This header will be checked with each API request to see if it is valid and not yet expired. Tokens currently expire after 8 hours.

### 54.3.1 Logout

Clients should use the logout endpoint to invalidate their token:

POST /api/v1/logout with the token in the HTTP Authorization header and you will get the following response upon successful invalidation of the API token:

```
{
  "message": "Your token has been revoked."
}
```

### 54.4 Errors

The API will respond to all errors (400-599) with a JSON object with the following fields:

```
{
  "message": "This is a detailed error message."
}
```

### 54.5 Endpoints

#### 54.5.1 Root Endpoint

Does not require authentication.

The root endpoint describes the available resources:

GET /api/v1/

Response 200 (application/json):

```
{
  "current_user_url": "/api/v1/user",
  "sources_url": "/api/v1/sources",
  "submissions_url": "/api/v1/submissions",
}
```
54.5.2 Sources [/sources]

Get all sources [GET]

Requires authentication. Provides a list of all sources and data about them (such as number of documents, submissions, and their public key that replies should be encrypted to).

GET /api/v1/sources

Response 200 (application/json):

```json
{
  "sources": [
    {
      "add_star_url": "/api/v1/sources/9b6df7c9-a6b1-461d-91f0-5b715fc7a47a/add_star",
      "interaction_count": 2,
      "is_flagged": false,
      "is_starred": false,
      "journalist_designation": "validated benefactress",
      "key": {
        "fingerprint": "8C71EA66B027309A31DBD691733DA6554DB12",
        "public": "-----BEGIN PGP PUBLIC KEY BLOCK-----
        mQINBFGRfoABEACf5Y+6prky4JcWmKSsuh/52ZLw1FTCqrgAIKOQVVFz+cY2r1FHv/njQXYB4bPOCTp7MybmMxxIKwXgJCAvKvKLbpi7p5X2Wkg+h+qGjgJjotqY9iPP6KQ3
        L/1Stn+ty4io3cDr7171S0tOmcoPKEVv6eGxSMcAYSmJnKKS1WScsfb28KhCzl/
        rBAp2qXt9uc6jQjhh1RPL3bGIG0cq37yX/zbFeefDBDF8m8d6nc7pvyvYMa09PGVibVg6h8h+rPq/
        rK7yYoW2l6MxW7cG/Ah+DkGxK8E8yuj92rNSPNQfqpAI/tX1dEcEv/54mba7uxrE33nsbnQFM2xTAIrSjXa904CEEWnRcznNE9u09FeM4b/b/1/Modsr7gOrWbO6QwctV/
        Yn7I7blUxZpMzDnsbgorR89aukS9VHG2nYL3zj0yWvRAiw3/CwsJEYajkIPacEdxXh2c8CCTTe8zXiPxxQOvynFBA666cfvq/5jsQhJ3NxRdVY/
        6W9dcCqXitcYnAnn00QKrkr81jXE3BEef7AQPXHIGsvFMImqeqUN+wQPGUkheomy7/7z3easKub3MYLkuW6y7c31z6cmvt6h5YcNPvQXCOx4 JBkVcK
        nPbzst612sbehqTQeEsDnVU1sPLpxfbkB5WuQ1EV8kjmd4j5
        3e54ntTbb/
        O\n/FVNVyyw5o9YzIQpsoYUKCcVCKv26b13gpfsDWjoyupKlh9AfFBBT3oiNhzeFhiv\n/25RbLGJYAuzWNdMDt3RAq2bzbFyQ
        RNwjeOqKpoG313mY vkRSRLz+N\n"bRF0A1G8lMOCrvv/9+eRtdrv+8fXlrqz55nGkt6J1i43Eiq1H0Eq7xwi+HMytDz/\nspWBR63INKhrK51Lm7yS0hBmnWuJpYeYv21hjzkkMbbfznD2zg
        P/CYsdnUbRHSfXmp16iigm\njoRfCc7J0j/SXkbyW9jyc+iTGCWfqqXa14u03Vn1YqFwqb+e/xa
        Qv14kwgsC17npn8O/
        X1y9gpV0qleikFgNts8sp+a21VdRSjh4Xh7I6THMgoUDJxtuE0evS3oBF2j\n/8OIDE8K21rJCY1xN+1/x1v2jAvfMvYT6kOXYxP3o3s2Hemsw2Pnj04ZMwTSyb\n/101xfgB1Xbd1hrv9Q5PNOwPxrh/7
        bf2QWtxup81u6T6qEerLiF0m/
        ShvYyKQsaJncLwUtW3Rj1BD5Cpd4+491j6cRXQ8xfYmC2d2MMbTtMAoq4DobYw75NKiss25gs6\n/6XNwCWOsR18QNBKkknpuh
        ----END PGP PUBLIC KEY BLOCK-----\n        "type": "PGP"
    },
    "last_updated": "2018-07-10T00:52:21.1574092",
    "number_of_documents": 0,
    "number_of_messages": 2,
    "remove_star_url": "/api/v1/sources/9b6df7c9-a6b1-461d-91f0-5b715fc7a47a/remove_star",
  ]
}
```

(continues on next page)
54.5.3 Individual Source [/sources/<source_uuid>]

Requires authentication.
An object representing a single source.

Response 200 (application/json):

```
{
    "add_star_url": "/api/v1/sources/9b6df7c9-a6b1-461d-91f0-5b715fc7a47a/add_star",
    "interaction_count": 2,
    "is_flagged": false,
    "is_starred": false,
    "journalist_designation": "validated benefactress",
    "key": {
        "fingerprint": "8C71EA66B027212F81F05B715FC7A47A",
        "type": "PGP"
    },
    "last_updated": "2018-07-10T00:52:21.157409Z",
    "number_of_documents": 0,
    "number_of_messages": 2,
    "remove_star_url": "/api/v1/sources/9b6df7c9-a6b1-461d-91f0-5b715fc7a47a/remove_star",
    "replies_url": "/api/v1/sources/9b6df7c9-a6b1-461d-91f0-5b715fc7a47a/replies",
    "submissions_url": "/api/v1/sources/9b6df7c9-a6b1-461d-91f0-5b715fc7a47a/submissions",
    "url": "/api/v1/sources/9b6df7c9-a6b1-461d-91f0-5b715fc7a47a",
    "uuid": "9b6df7c9-a6b1-461d-91f0-5b715fc7a47a"
}
```

Get all submissions associated with a source [GET]

Requires authentication.

GET /api/v1/sources/<source_uuid>/submissions

Response 200 (application/json)
Get a single submission associated with a source [GET]

Requires authentication.

GET /api/v1/sources/<source_uuid>/submissions/<submission_uuid>

Response 200 (application/json):

```json
{
   "download_url": "/api/v1/sources/598b859c-72c7-4e53-a68c-b725eb514241/submissions/4c2e701c-70d2-4cb5-87c0-de59c2ebbc62/download",
   "filename": "1-dejected_respondent-msg.gpg",
   "is_read": false,
   "size": 603,
   "source_url": "/api/v1/sources/598b859c-72c7-4e53-a68c-b725eb514241",
   "submission_url": "/api/v1/sources/598b859c-72c7-4e53-a68c-b725eb514241/submissions/4c2e701c-70d2-4cb5-87c0-de59c2ebbc62",
   "uuid": "4c2e701c-70d2-4cb5-87c0-de59c2ebbc62"
}
```

Get all replies associated with a source [GET]

Requires authentication.

GET /api/v1/sources/<source_uuid>/replies

Response 200 (application/json):
Get a single reply associated with a source [GET]

Requires authentication.

GET /api/v1/sources/<source_uuid>/replies/<reply_uuid>

Response 200 (application/json):

```json
{
    "filename": "3-famished_sheep-reply.gpg",
    "is_deleted_by_source": false,
    "journalist_username": "journalist",
    "journalist_first_name": "Bob",
    "journalist_last_name": "Smith",
    "journalist_uuid": "a2405127-1c9e-4a3a-80ea-95f6a7e5738",
    "reply_url": "/api/v1/sources/f381dbb4-4bb5-451a-801a-e961461af6e5/replies/98cc4ed6-6ac5-4867-b144-f97d0497f2c1",
    "size": 1116,
    "source_url": "/api/v1/sources/f381dbb4-4bb5-451a-801a-e961461af6e5",
    "uuid": "98cc4ed6-6ac5-4867-b144-f97d0497f2c1"
}
```
Download a reply [GET]

Requires authentication.

GET /api/v1/sources/<source_uuid>/replies/<reply_uuid>/download

Response 200 will have Content-Type: application/pgp-encrypted and is the content of the PGP encrypted reply.

An ETag header is also present containing the SHA256 hash of the response data:

"sha256:c757c5aa263dc4a5a2bca8e7fe973367dbd2c1a6c780d19c0ba499e6b1b81efa"

Note that these are not intended for cryptographic purposes and are present for clients to check that downloads are not corrupted.

Delete a reply [DELETE]

Requires authentication.

DELETE /api/v1/sources/<source_uuid>/replies/<reply_uuid>

Response 200:

{  "message": "Reply deleted" }

Add a reply to a source [POST]

Requires authentication. Clients are expected to encrypt replies prior to submission to the server. Replies should be encrypted to the public key of the source.

Including the uuid field in the request is optional. Clients may want to pre-set the uuid so they can track in-flight messages.

POST /api/v1/sources/<source_uuid>/replies

with the reply in the request body:

{  "uuid": "0bc588dd-f613-4999-b21e-1cebbd9adc2c",  "reply": "-----BEGIN PGP MESSAGE-----[...]-END PGP MESSAGE-----" }

Response 201 created (application/json):

{  "message": "Your reply has been stored",  "uuid": "0bc588dd-f613-4999-b21e-1cebbd9adc2c" }

The returned uuid field is the UUID of the reply and can be used to reference this reply later. If the client set the uuid in the request, this will have the same value.

Replies that do not contain a GPG encrypted message will be rejected:
Response 400 (application/json):

```json
{
    "message": "You must encrypt replies client side"
}
```

### Delete a submission [DELETE]

Requires authentication.

```
DELETE /api/v1/sources/<source_uuid>/submissions/<submission_uuid>
```

Response 200:

```json
{
    "message": "Submission deleted"
}
```

### Download a submission [GET]

Requires authentication.

```
GET /api/v1/sources/<source_uuid>/submissions/<submission_uuid>/download
```

Response 200 will have Content-Type: application/pgp-encrypted and is the content of the PGP encrypted submission.

An ETag header is also present containing the SHA256 hash of the response data:

```
"sha256:c757c5aa263dc4a5a2bca8e7fe973367dbd2c1a6c780d19c0ba499e6b1b81efa"
```

Note that these are not intended for cryptographic purposes and are present for clients to check that downloads are not corrupted.

### Delete a Source and all their associated submissions [DELETE]

Requires authentication.

```
DELETE /api/v1/sources/<source_uuid>
```

Response 200:

```json
{
    "message": "Source and submissions deleted"
}
```

### Star a source [POST]

Requires authentication.

```
POST /api/v1/sources/<source_uuid>/star
```

Response 201 created:
Remove a source [DELETE]

Requires authentication.

DELETE /api/v1/sources/<source_uuid>/star

Response 200:

```json
{
  "message": "Star removed"
}
```

Flag a source [POST]

Requires authentication.

POST /api/v1/sources/<source_uuid>/flag

Response 200:

```json
{
  "message": "Source flagged for reply"
}
```

54.5.4 Submission [/submissions]

Get all submissions [GET]

Requires authentication. This gets details of all submissions across sources.

GET /api/v1/submissions

Response 200:

```json
{
  "submissions": [
  {
    "download_url": "/api/v1/sources/1ed4c191-c6b1-463b-92a5-102deaf7d40a/
→submissions/e58f6206-fc12-4dbe-9a9c-84c3d82eea2f/download",
    "filename": "1-abridged_psalmist-msg.gpg",
    "is_read": false,
    "size": 604,
    "source_url": "/api/v1/sources/1ed4c191-c6b1-463b-92a5-102deaf7d40a",
    "submission_url": "/api/v1/sources/1ed4c191-c6b1-463b-92a5-102deaf7d40a/
→submissions/e58f6206-fc12-4dbe-9a9c-84c3d82eea2f",
    "uuid": "e58f6206-fc12-4dbe-9a9c-84c3d82eea2f"
  },
  {
    "download_url": "",
    "message": "errordata"
  }
  ]
}
```
54.5.5 Reply [/replies]

Get all replies [GET]

Requires authentication. This gets details of all replies across sources.

GET /api/v1/replies

Response 200:

```json
{
  "replies": [
    {
      "filename": "3-famished_sheep-reply.gpg",
      "is_deleted_by_source": false,
      "journalist_username": "journalist",
      "journalist_uuid": "a2405127-1c9e-4a3a-80ea-95f6a7e5738",
      "journalist_first_name": "Bob",
      "journalist_last_name": "Smith",
      
```
54.5.6 User [/user]

Get an object representing the current user [GET]

Requires authentication.
GET /api/v1/user

Response 200:

```json
{
  "is_admin": true,
  "last_login": "2018-07-09T20:29:41.696782Z",
  "username": "journalist",
  "uuid": "a2405127-1c9e-4a3a-80ea-95f6a71e5738",
  "first_name": "Bob",
  "last_name": "Smith",
}
```
SecureDrop is a multi-server system, and you may need the full server stack available in order to develop and test some features. To make this easier, the project includes a Vagrantfile that can be used to create two predefined virtual environments:

- **Staging**
- **Production**

This document explains the purpose of, and how to get started working with, each one.

Note: If you plan to alter the configuration of any of these machines, make sure to review the *Testing: Configuration Tests* documentation.

Note: If you see test failures due to *Too many levels of symbolic links* and you are using VirtualBox, try restarting VirtualBox.

### 55.1 Staging

A compromise between the development and production environments. This configuration can be thought of as identical to the production environment, with a few exceptions:

- The Debian packages are built from your local copy of the code, instead of installing the current stable release packages from `https://apt.freedom.press`.

- The staging environment is configured for direct SSH access so it’s more ergonomic for developers to interact with the system during debugging.

- The Postfix service is disabled, so OSSEC alerts will not be sent via email.

This is a convenient environment to test how changes work across the full stack.

You should first bring up the VM required for building the app code Debian packages on the staging machines:
To rebuild the local packages for the app code and update on Xenial staging:

```
make build-debs
make staging
```

The Debian packages will be rebuilt from the current state of your local git repository and then installed on the staging servers.

Note: If you are using macOS and you run into errors from Ansible such as OSError: [Errno 24] Too many open files, you may need to increase the maximum number of open files. Some guides online suggest a procedure to do this that involves booting to recovery mode and turning off System Integrity Protection (`csrutil disable`). However this is a critical security feature and should not be disabled. Instead follow this procedure to increase the file limit.

Set `/Library/LaunchDaemons/limit.maxfiles.plist` to the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Label</key>
    <string>limit.maxfiles</string>
    <key>ProgramArguments</key>
    <array>
      <string>launchctl</string>
      <string>limit</string>
      <string>maxfiles</string>
      <string>65536</string>
      <string>65536</string>
    </array>
    <key>RunAtLoad</key>
    <true/>
    <key>ServiceIPC</key>
    <false/>
  </dict>
</plist>
```

The plist file should be owned by `root:wheel`:

```
sudo chown root:wheel /Library/LaunchDaemons/limit.maxfiles.plist
```

This will increase the maximum open file limits system wide on macOS (last tested on 10.11.6).

The web interfaces and SSH are available over Tor. A copy of the the Onion URLs for Source and Journalist Interfaces,
as well as SSH access, are written to the Vagrant host’s install_files/ansible-base directory.

To access the Source Interface from Tor Browser, use the v3 onion URL from the file install_files/ansible-base/app-sourcev3-ths.

To use the Journalist Interface, you will need to modify Tor Browser’s configuration to allow access to an authenticated onion service:

- First, add the following line to your Tor Browser’s torrc file, typically found at tor-browser_en-US/Browser/TorBrowser/Data/Tor/torrc:

  ```
  ClientOnionAuthDir TorBrowser/Data/Tor/onion_auth
  ```

- Next, create the onion_auth directory:

  ```
  mkdir tor-browser_en-US/Browser/TorBrowser/Data/Tor/onion_auth
  chmod 0700 tor-browser_en-US/Browser/TorBrowser/Data/Tor/onion_auth
  ```

- Finally, copy the file install_files/ansible-base/app-journalist.auth_private to the onion_auth directory and restart Tor Browser. You should now be able to visit the v3 onion address in install_files/ansible-base/app-journalist.auth_private from Tor Browser.

For working on OSSEC monitoring rules with most system hardening active, update the OSSEC-related configuration in install_files/ansible-base/staging.yml so you receive the OSSEC alert emails.

Direct SSH access is available for staging hosts, so you can use molecule login -s <scenario> -h app-staging, where <scenario> is either virtualbox-staging-xenial or libvirt-staging-xenial, depending on your environment.

**55.2 Production**

This is a production installation with all of the system hardening active, but virtualized, rather than running on hardware. You will need to configure prod-like secrets, or export ANSIBLE_ARGS="--skip-tags validate" to skip the tasks that prevent the prod playbook from running with Vagrant-specific info.

You can provision production VMs from an Admin Workstation (most realistic), or from your host. If your host OS is Linux-based and you plan to use an Admin Workstation, you will need to switch Vagrant’s default virtualization provider from virtualbox to libvirt. The Admin Workstation VM configuration under Linux uses QEMU/KVM, which cannot run simultaneously with Virtualbox.

Instructions for both installation methods follow.

**55.2.1 Switching to the Vagrant libvirt provider**

Make sure you’ve already installed Vagrant, as described in the multi-machine setup docs.

**Ubuntu 16.04 setup**

Install libvirt and QEMU:

```
sudo apt-get update
sudo apt-get install libvirt-bin libvirt-dev qemu-utils qemu virt-manager
sudo /etc/init.d/libvirt-bin restart
```

Add your user to the libvirtd group:
sudo addgroup libvirtd
sudo usermod -a -g libvirtd $USER

Install the required Vagrant plugins for converting and using libvirt boxes:

vagrant plugin install vagrant-libvirt
vagrant plugin install vagrant-mutate

**Note:** If Vagrant is already installed it may not recognize libvirt as a valid provider. In this case, remove Vagrant with

```
sudo apt-get remove vagrant
```
and reinstall it.

Log out, then log in again. Verify that libvirt is installed and KVM is available:

```
libvirtd --version
kvm-ok
```

**Debian 9 setup**

Install Vagrant, libvirt, QEMU, and their dependencies:

```
sudo apt-get update
sudo apt-get install -y vagrant vagrant-libvirt libvirt-daemon-system qemu-kvm virt-\manager
sudo apt-get install -y ansible rsync
vagrant plugin install vagrant-libvirt
vagrant plugin install vagrant-mutate
sudo usermod -a -G libvirt $USER
sudo systemctl restart libvirtd
```

Add your user to the kvm group to give it permission to run KVM:

```
sudo usermod -a -G kvm $USER
sudo rmmod kvm_intel
sudo rmmod kvm
sudo modprobe kvm
sudo modprobe kvm_intel
```

Log out, then log in again. Verify that libvirt is installed and your system supports KVM:

```
sudo libvirtd --version
[ `egrep -c 'flags\s*:.*(vmx|svm)' /proc/cpuinfo` -gt 0 ] && 
  echo "KVM supported!" || echo "KVM not supported..."
```

**Set libvirt as the default provider**

Set the default Vagrant provider to libvirt:

```
echo 'export VAGRANT_DEFAULT_PROVIDER=libvirt' >> ~/.bashrc
export VAGRANT_DEFAULT_PROVIDER=libvirt
```
Note: To explicitly specify the libvirt provider below, use the command `vagrant up --provider=libvirt /prod/`

Convert Vagrant boxes to libvirt

Convert the VirtualBox images for Xenial from `virtualbox` to `libvirt` format:

```
vagrant box add --provider virtualbox bento/ubuntu-16.04
vagrant mutate bento/ubuntu-16.04 libvirt
```

You can now use the libvirt-backed VM images to develop against the SecureDrop multi-machine environment.

55.2.2 Install from an Admin Workstation VM

In SecureDrop, admin tasks are performed from a Tails Admin Workstation. You should configure a Tails VM in order to install the SecureDrop production VMs by following the instructions in the Virtualizing Tails guide.

Once you’re prepared the Admin Workstation, you can start each VM:

```
vagrant up --no-provision /prod/
```

At this point you should be able to SSH into both `app-prod` and `mon-prod`. From here you can follow the server configuration instructions to test connectivity and prepare the servers. These instructions will have you generate SSH keys and use `ssh-copy-id` to transfer the key onto the servers.

Note: If you have trouble SSHing to the servers from Ansible, remember to remove any old ATHS files in `install_files/ansible-base`.

Now from your Admin workstation:

```
cd ~/Persistent/securedrop
./securedrop-admin setup
./securedrop-admin sdconfig
./securedrop-admin install
```

Note: The sudo password for the `app-prod` and `mon-prod` servers is by default `vagrant`.

After install you can configure your Admin Workstation to SSH into each VM via:

```
./securedrop-admin tailsconfig
```

55.2.3 Install from Host OS

If you are not virtualizing Tails, you can manually modify `site-specific`, and then provision the machines. You should set the following options in `site-specific`:
Note that you will also need to generate Submission and OSSEC PGP public keys, and provide email credentials to send emails to. Refer to this document on configuring prod-like secrets for more details on those steps.

To create the prod servers, run:

```bash
vagrant up /prod/
vagrant ssh app-prod
sudo -u www-data bash
cd /var/www/securedrop/
./manage.py add-admin
```

A copy of the Onion URLs for Source and Journalist Interfaces, as well as SSH access, are written to the Vagrant host's `install_files/ansible-base` directory, named:

- app-sourcev3-ths
- app-journalist.auth_private
- app-ssh.auth_private
- mon-ssh.auth_private

### 55.2.4 SSH Access

By default, direct SSH access is not enabled in the prod environment. You will need to log in over Tor after initial provisioning or set `enable_ssh_over_tor` to “false” during `./securedrop-admin tailsconfig`. See Connecting to VMs via SSH Over Tor or Configuring SSH for Local Access for more info.
Virtual Environments: Admin Workstation

SecureDrop uses Tails for the Admin Workstation environment. In order to perform a fully virtualized production install, you will need to first set up Tails in a virtual machine.

**Note:** For the instructions that follow, you need to download the most recent Tails ISO from the Tails website.

### 56.1 macOS

For the macOS instructions, you will use VirtualBox to create a Tails VM that you can use to install SecureDrop on app-prod and mon-prod.

#### 56.1.1 Create a VirtualBox VM

1. Open VirtualBox
2. Click **New** to create a new VM with the following options:
   - **Name:** “Admin Workstation”
   - **Type:** “Linux”
   - **Version:** “Debian (64-bit)”

**Note:** You may call the VM a different name, but you must replace “Admin Workstation” later on in these instructions with the name you select.

3. Click **Continue**.
4. At the prompt, configure at least 2048 MB of RAM. Click **Continue**.
5. Leave the default Create a virtual hard disk now selected and click Create. All the default options (Hard disk file type: VDI (VirtualBox Disk Image) and Dynamically allocated) are fine. Click Create.

56.1.2 Booting Tails

Now that the VM is set up, you are ready to boot to Tails. Select the new VM in the VirtualBox sidebar, and click Settings.

1. Click Storage.
2. Click Empty under Controller: IDE.
3. Click the CD icon next to Optical Drive: and click Choose Virtual Optical Disk File.
4. Navigate to the Tails ISO to boot from.
5. Click General then Advanced.
6. Under Shared Clipboard select Bidirectional instead of Disabled. This option will enable you to transfer text from your host to the Tails VM, which we will use later on in these steps.

Note: Alternatively you can open these docs in Tor Browser in Tails. This will obviate the need to copy and paste between the guest and host OS.

56.1.3 Install Tails

Next you will install Tails onto the Virtual Hard Disk Image. Start the VM, boot to Tails, and enter an administration password and start Tails.

Note: For all the instructions that follow, you will need to configure an administration password each time you boot Tails.

1. Copy the following patch and save it as installer.patch in a folder in your Tails VM:

```plaintext
--- /usr/lib/python2.7/dist-packages/tails_installer/creator.py 2018-01-22 14:59:40 2018-03-05 05:15:00
+++ /usr/lib/python2.7/dist-packages/tails_installer/creator.py.mod 2018-03-05 05:15:00
@@ -595,16 +595,6 @@
class LinuxTailsInstallerCreator(TailsInstallerCreator):
    self.log.debug('Skipping non-removable device: %s' % data['device'])
-
-    # Only pay attention to USB and SDIO devices, unless --force'd
-    iface = drive.props.connection_bus
-    if iface != 'usb' and iface != 'sdio' 
-        and self.opts.force != data['device']:
-            self.log.warning("Skipping device '%%(device)s' connected to '%%(interface)s'
-            interface" % {'device': data['udi'], 'interface': iface})
-            continue
-
-    # Skip optical drives
```

(continues on next page)
2. Now run the following two commands in a Terminal in your Tails VM:

```bash
sudo patch -p0 -d/ < installer.patch
sudo /usr/bin/python -tt /usr/bin/tails-installer -u -n --clone -P -m -x
```

3. The Tails Installer will appear. Click Install Tails.

4. Once complete, navigate to Applications, Utilities and open Disks.

5. Click on the disk named “Tails” and click the Play icon to mount the disk.

6. Next open /media/amnesia/Tails/syslinux/live*.cfg and delete all instances of
   live-media=removable.

7. Shut down the VM.

### 56.1.4 Boot to Tails Hard Drive Install

Now we will remove the CD and boot to the Tails we just installed on our virtual hard drive. From macOS you should:

1. Click the VM in the sidebar of VirtualBox and click Settings.
2. Click Storage and select the Tails .iso under Controller: IDE.
3. Click the CD icon, then Remove Disk from Virtual Drive.
4. Click Ok.
5. Start the VM.

### 56.1.5 Configure Persistence

Now in your booted Tails VM you should:
1. Configure an admin password when prompted.

2. Copy the following patch to the Tails VM and save it as `persistence.patch`:

```diff
--- /usr/share/perl5/Tails/Persistence/Setup.pm 2017-06-30 09:56:25.000000000
+++ /usr/share/perl5/Tails/Persistence/Setup.pm.mod 2017-07-20 07:17:48.472000000
@@ -404,19 +404,6 @@
    my @checks = (         
        {               
            method => 'drive_is_connected_via_a_supported_interface',
            message => $self->encoding->decode(gettext("Tails is running from non-USB / non-SDIO device %s.")),
            needs_drive_arg => 1,
        },
        {               
            method => 'drive_is_optical',
            message => $self->encoding->decode(gettext("Device %s is optical.")),
            must_be_false => 1,
            needs_drive_arg => 1,
        },
        {               
            method => 'started_from_device_installed_with_tails_installer',
            message => $self->encoding->decode(gettext("Device %s was not created using Tails Installer.")),
        },
    )
```

3. To apply the patch, from the Terminal run:

```
sudo patch -p0 -d/ < persistence.patch
```

4. Navigate to **Applications** then **Tails** and click **Configure persistent volume**. Configure a persistent volume enabling all persistence options.

**56.1.6 Shared Folders**

1. In macOS, click on the Tails VM in VirtualBox and then go to **Settings**.

2. Click on **Shared Folders** and click the button on the right hand side to add the folder. Navigate to the location of the SecureDrop repository on your local machine. Check **Auto-mount**. Do not check **Read-only**.

3. Now reboot your Tails VM, decrypt the Persistent volume, and run the following commands in a **Terminal** in Tails:

```bash
mkdir ~/Persistent/securedrop
echo 'if [ ! -d ~/Persistent/securedrop/install_files ]; then sudo mount -t vboxsf -o _uid=$UID,_gid=$(id -g) securedrop ~/Persistent/securedrop; fi' >> /live/persistence/...TailsData_unlocked/dotfiles/.bashrc
```

The first time you open a Terminal in that session you will be prompted for your sudo password and the shared folder will be mounted. Each time you open a Terminal thereafter in the Tails session, your sudo password will not be needed.

**56.1.7 Allow the Guest to Create Symlinks**

Finally, you’ll need to allow the guest to create symlinks, which are **disabled by default** in VirtualBox.
Shut down the Tails VM, and in your host run:

```
VBoxManage setextradata "Admin Workstation" VBoxInternal2/→SharedFoldersEnableSymlinksCreate/securedrop 1
```

**Note:** If you named your Tails VM something other than “Admin Workstation”, you can run `VBoxManage list vms` to get the name of the Virtual Machine.

Finally, restart VirtualBox.

### 56.1.8 Configure Networking

In order to communicate with the server VMs, you’ll need to attach this virtualized Admin Workstation to the securedrop network.

**Warning:** If you named the SecureDrop repository something other than securedrop, you should connect your VM to the network of the same name.

With the Admin Workstation VM turned off, you should:

1. Click on the VM in VirtualBox.
2. Click **Settings**.
3. Click **Network** and then **Adapter 2**.
4. Enable this network adapter and attach it to the **Internal Network** called securedrop.
5. Click OK and start the VM.

Now you should be able to boot to Tails, decrypt the Persistent volume, navigate to `~/Persistent/securedrop` and proceed with the **production install**.

### 56.1.9 Disable Shared Clipboard (Optional)

1. Click on the VM in VirtualBox.
2. Click **Settings**.
3. Click **General** and then **Advanced**.
4. Now that you are finished with copy pasting the patches above you can change the **Shared Clipboard** from **Bidirectional** back to **Disabled**.

### 56.2 Linux

For the Linux instructions, you will use KVM/libvirt to create a Tails VM that you can use to install SecureDrop on app-prod and mon-prod.
56.2.1 Create a libvirt VM

Follow the Tails virt-manager instructions for Running Tails from a virtual USB storage. After installing Tails on the removable USB device, shut down the VM and edit the boot options. You’ll need to manually enable booting from the USB device by checking the box labeled USB Disk 1.

Then proceed with booting to the USB drive, and configure a persistence volume.

56.2.2 Shared Folders

In order to mount the SecureDrop git repository as a folder inside the Tails persistence volume, you must add a filesystem via virt-manager.

1. Choose View Details to edit the configuration of the virtual machine.
2. Click on the Add Hardware button on the bottom of the left pane.
3. Select Filesystem in the left pane.
4. In the right pane, change the Mode to Mapped.
5. In the right pane, change Source path to the path to the SecureDrop git repository on the host machine.
6. In the right pane, change **Target path** to **securedrop**.

7. Click **Finish**.

On the next VM boot, you will be able to mount the SecureDrop git repository from the host machine via:

```bash
mkdir -p ~/Persistent/securedrop
sudo mount -t 9p securedrop ~/Persistent/securedrop
```

You will need to run the `mount` command every time you boot the VM. By default only read operations are supported. In order to support modifying files in the git repository, you will need to configure file ACLs. On the host machine, from within the SecureDrop git repository, run:
make libvirt-share

All files will be created with mode 0600 and ownership libvirt-qemu:libvirt-qemu. You will need to modify the files manually on the host machine in order to commit them.
The SecureDrop project includes Molecule scenarios for developing and testing against multi-server configurations, including a scenario to simulate the process of upgrading an existing system. This document explains how to work with this scenario to test features that make potentially release-breaking changes such as database schema updates.

The Molecule upgrade scenario sets up a predefined staging Securedrop virtual environment using Vagrant boxes built with the latest application release. It also creates a virtualized APT repository, and modifies the SecureDrop environment to use this APT repository instead of the FPF main repo at https://apt.freedom.press/.

You can use this scenario to test the upgrade process, using either locally-built .debs or packages from the FPF test repo at https://apt-test.freedom.press/. Both options are described below.

**Note:** The upgrade scenario uses QEMU/KVM via Vagrant’s libvirt provider, in place of the default Virtualbox provider. If you haven’t already done so, you’ll need to set up the libvirt provider before proceeding. For more information, see *Switching to the Vagrant libvirt provider.*

### 57.1 Upgrade testing using locally-built packages

**Note:** As of 0.12.1, the default platform for upgrade testing boxes is Ubuntu Xenial 16.04. We no longer support upgrade boxes based on Ubuntu Trusty 14.04.

First, build the app code packages and create the environment:

```bash
make build-debs
make upgrade-start
```

The playbook will return the source interface Onion address. You can use this to check the application version displayed in the source interface footer. Alternatively, you can log into the *Application Server* VM and check the deployed package version directly:
molecule login -s upgrade -h app-staging

From the Application Server:

apt-cache policy securedrop-config

The installed package version should match the latest release version.

To perform an upgrade using the virtualized APT repository, log out of the Application Server and run the Molecule side-effect action:

make upgrade-test-local

This will upgrade the SecureDrop packages on the Application and Monitor Servers, using your locally-built packages and apt VM instead of the FPF production apt repository.

You can verify that the application version has changed either by checking the source interface’s footer or directly on the Application Server as described above.

57.2 Upgrade testing using apt-test.freedom.press

You can use the upgrade scenario to test upgrades using official release candidate packages from the FPF test APT repository. First, create the environment:

make upgrade-start-qa

Then, log into the Application Server:

molecule login -s upgrade -h app-staging

From the Application Server:

sudo apt-get update
apt-cache policy securedrop-config

The installed package version should match the current release version. To install the latest packages from the apt-test proxy:

make upgrade-test-qa

Log back into the Application Server, and repeat the previous commands:

sudo apt-get update
apt-cache policy securedrop-config

Navigate to the Source Interface URL again, and confirm you see the upgraded version in the footer. Then proceed with testing the new version.

57.3 Updating the base boxes used for upgrade testing

When a new version of SecureDrop is released, we must create and upload new VM images, to enable testing against that base version in future upgrade testing. The procedure is as follows:

1. make clean to remove any previous artifacts (which would also be pushed)
2. `git checkout <version>` (if a point release, `git checkout develop`)
3. `make vagrant-package`
4. `cd molecule/vagrant-packager && ./push.yml` to upload to S3
5. Commit the local changes to JSON files and open a PR.

Subsequent invocations of `make upgrade-start` will pull the latest version of the box.
Contributing Guidelines

58.1 Signing commits

Commits should be signed, as explained in the GitHub documentation. This helps verify commits proposed in a pull request are from the expected author.

58.2 Branching Strategy

Development for the upcoming release of SecureDrop takes place on `develop`, which is the default branch. If you want to contribute, you should branch from and submit pull requests to `develop`. If you want to install or audit SecureDrop, you should use the latest tag that is not a release candidate (e.g. `0.6` not `0.6-rc1`).

Tip: After you have cloned the SecureDrop repository, you can run `git tag` locally to see all the tags. Alternatively, you can view them on GitHub.

58.3 Automated Testing

When a pull request is submitted, we have Circle CI automatically run the SecureDrop test suites, which consist of:

1. Unit tests of the Python SecureDrop application code.
2. Functional tests that use Selenium to drive a web browser to verify the function of the application from the user’s perspective.
3. Tests of the system configuration state using testinfra.

Before a PR can be merged, these tests must all pass. If you modify the application code, you should verify the tests pass locally before submitting your PR. If you modify the server configuration, you should run the testinfra tests. Please denote in the checklist when you submit the PR that you have performed these checks locally.
58.4 Code Style

We use code linters to keep a consistent code quality and style. These linters also run in CI and will produce build failures. To avoid this, we have included a git pre-commit hook. You can install it with the following command run at the root of the repository:

```
l -sf ../../../git/pre-commit .git/hooks/pre-commit
```

**Note:** The code linters are installed automatically on the Development VM, but for the pre-commit hook to work, you will need to install the linting tools locally on your host machine. From the root of the repo you can run the following:

```
pip install -r securedrop/requirements/develop-requirements.txt
```

58.4.1 Python

All Python code should be `flake8` compliant. You can run `flake8` locally via:

```
make flake8
```

58.4.2 Shell

All Shell code (e.g. `bash`, `sh`) should be `shellcheck` compliant. You can run `shellcheck` locally via:

```
make shellcheck
```

For reference, consult the shellcheck wiki for detailed explanations of any reported violations.

58.4.3 HTML

HTML should be in compliance with Google’s HTML style guide. We use html-linter to lint our HTML templates in `securedrop/source_templates` and `securedrop/journalist_templates`. Run the HTML linting options we use via:

```
make html-lint
```

58.4.4 YAML

The Ansible configuration is specified in YAML files, including variables, tasks, and playbooks. All YAML files in the project should pass the `yamllint` standards declared in the `.yamllint` file at the root of the repository. Run the checks locally via:

```
make yamllint
```
58.5 Type Hints in Python code

By adding type hints/annotations in the Python code, we are making the codebase easier to maintain in the long run by explicitly specifying the expected input/output types of various functions.

Any pull request with Python code in SecureDrop should have corresponding type hints for all the functions. Type hints and function annotations are defined in PEP 484 and in PEP 3107. We also use the mypy tool in our CI to find bugs in our Python code.

If you are new to Python type hinting, please read the above mentioned PEP documents, and then go through the examples in mypy documentation. SecureDrop uses Python2, that is why all type annotation/hints are mentioned as code comments. The typing module is also not part of the standard library. It is being installed as part of the development and run-time dependency.

58.5.1 Example of Type Hint

```python
import typing
# https://www.python.org/dev/peps/pep-0484/#runtime-or-type-checking
if typing.TYPE_CHECKING:
    # flake8 can not understand type annotation yet.
    # That is why all type annotation relative import
    # statements has to be marked as noqa.
    from typing import Dict  # noqa: F401

class Config(object):
    def __init__(self):
        # type: () -> None
        self.NAMES = {}  # type: Dict[str, str]

    def add(self, a, b):
        # type: (int, int) -> float
        c = 10.5  # type: float
        return a + b + c

    def update(self, uid, Name):
        # type: (int, str) -> None
        """
        This method updates the name example.
        """
        self.NAMES[uid] = Name

    def main():
        # type: () -> None
        config = Config()  # type: Config
        config.add(2, 3)
        config.update(223, "SD")

if __name__ == '__main__':
    main()
```

The above example shows how to do a conditional import of Dict class from typing module. typing.TYPE_CHECKING will only be true when we use mypy to check type annotations.
58.5.2 How to Use mypy?

`make lint` already checks for any error using the `mypy` tool. In case you want to have a local installation, you can do that using a Python3 virtualenv.

```
$ python3 -m venv ../.py3
$ source ../.py3/bin/activate
$ pip install mypy
$ mypy --py2 securedrop
```

58.6 Git History

We currently use an explicit merge strategy to merge feature branches into `develop`. In order to keep our git history as clean as possible, please squash your commits to package up your changes into a clear history. If you have many unnecessary commits that do not add information to aid in review, they should be removed. If you are unfamiliar with how to squash commits with rebase, check out this blog post.

58.7 Privileges

**Note:** The privilege escalation workflow is different for code maintainers and translation maintainers.

Dedicated contributors to SecureDrop will be granted extra privileges such as the right to push new branches or to merge pull requests. Any contributor with the right technical and social skills is entitled to ask. The people who have the power to grant such privileges are committed to do so in a transparent way as follows:

1. The contributor posts a message in the forum asking for privileges (review or merge, etc.).
2. After at least a week someone with permissions to grant such privilege reviews the thread and either:
   - grants the privilege if there are no objections from current maintainers and adds a message to the thread; or
   - explains what is expected from the contributor before they can be granted the privilege.
3. The thread is closed.

The privileges of a developer who has not been active for six months or more are revoked. They can apply again at any time.

58.8 Other Tips

- To aid in review, please write clear commit messages and include a descriptive PR summary. We have a PR template that specifies the type of information you should include.
- To maximize the chance that your PR is merged, please include the minimal changes to implement the feature or fix the bug.
- If there is not an existing issue for the PR you are interested in submitting, you should submit an issue first or comment on an existing issue outlining how you intend to approach the problem.
59.1 Using Tor Browser with the Development Environment

We strongly encourage sources to use the Tor Browser when they access the Source Interface. Tor Browser is the easiest way for the average person to use Tor without making potentially catastrophic mistakes, makes disabling JavaScript easy via the handy NoScript icon in the toolbar, and prevents state about the source’s browsing habits (including their use of SecureDrop) from being persisted to disk.

Since Tor Browser is based on an older version of Firefox (usually the current ESR release), it does not always render HTML/CSS the same as other browsers (especially more recent versions of browsers). Therefore, we recommend testing all changes to the web application in the Tor Browser instead of whatever browser you normally use for web development. Unfortunately, it is not possible to access the local development servers by default, due to Tor Browser’s proxy configuration.

To test the development environment in Tor Browser, you need to add an exception to allow Tor Browser to access localhost:

1. Open the “Tor Browser” menu and click “Preferences…”
2. Choose the “Advanced” section and the “Network” subtab under it
3. In the “Connection” section, click “Settings…”
4. In the text box labeled “No Proxy for:”, enter 127.0.0.1
   • Note: for some reason, localhost doesn’t work here.
5. Click “Ok” and close the Preferences window

You should now be able to access the development server in the Tor Browser by navigating to 127.0.0.1:8080 and 127.0.0.1:8081.
59.2 Upgrading or Adding Python Dependencies

We use a pip-compile based workflow for adding Python dependencies. If you would like to add a Python dependency, instead of editing the securedrop/requirements/*.txt files directly, please:

1. Edit the relevant *.in file in securedrop/requirements/
2. Use the following shell script to generate securedrop/requirements/*.txt files:
   ```bash
   make update-pip-dependencies
   ```
3. Commit both the securedrop/requirements/*.in and securedrop/requirements/*.txt files

59.3 Connecting to VMs via SSH Over Tor

59.3.1 Ubuntu/Debian Setup

You will need to install a specific variant of the nc tool in order to support the -x option for specifying a proxy host. macOS already runs the OpenBSD variant by default.

```
sudo apt-get install netcat-openbsd
```

After installing netcat-openbsd and appending the Tor config options to your local torrc, you can export the environment variable SECUREDROP_SSH_OVER_TOR=1 in order to use vagrant ssh to access the staging or prod instances over Tor. Here is an example of how that works:

```
$ vagrant up --provision /prod/  # restricts SSH to Tor after final reboot
$ vagrant ssh-config app-prod    # will show incorrect info due to lack of env var
Host app-prod
  HostName 127.0.0.1
  User vagrant
  Port 2201
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile /home/conor/.vagrant.d/insecure_private_key
  IdentitiesOnly yes
  LogLevel FATAL

$ vagrant ssh app-prod -c 'echo hello'  # will fail due to incorrect ssh-config
ssh_exchange_identification: read: Connection reset by peer

$ export SECUREDROP_SSH_OVER_TOR=1    # instruct Vagrant to use Tor for SSH
$ vagrant ssh-config app-prod          # will show correct info, with ProxyCommand
Host app-prod
  HostName 157xqhqltlu323vi.onion
  User vagrant
  Port 22
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile /home/conor/.vagrant.d/insecure_private_key
  IdentitiesOnly yes
  LogLevel FATAL
```

(continues on next page)
ProxyCommand nc -x 127.0.0.1:9050 %h %p

$ # ensure ATHS values are active in local Tor config:
$ cat *-aths | sudo tee -a /etc/tor/torrc > /dev/null && sudo service tor reload
$ vagrant ssh app-prod -c 'echo hello'  # works
hello
Connection to l57xhqhltlu323vi.onion closed.

If `SECUREDROP_SSH_OVER_TOR` is true, Vagrant will look up the ATHS URLs for each server by examining the contents of `app-ssh-aths` and `mon-ssh-aths` in `./install_files/ansible-base`. You can manually inspect these files to append values to your local `torrc`, as in the `cat` example above. Note that the `cat` example above will also add the ATHS info for the Journalist Interface, as well, which is useful for testing.

**Note:** The instructions above refer to VMs set up with v2 onion services. If v3 onion services are configured instead, the steps required for the local `tor` setup will differ. You will need to add a `ClientOnionAuthDir` directive to `torrc`, pointing to a directory containing the `*.auth_private` files created during the installation process under `install_files/ansible-base`.

### 59.4 Architecture Diagrams

Some helpful diagrams for getting a sense of the SecureDrop application architecture are stored here, including a high-level view of the SecureDrop database structure:
SecureDrop uses Alembic for database schema migrations. This guide is not a complete explanation of what alembic is or how it is used, so the original documentation should be read.

### 60.1 Migration Files

In the securedrop/ directory, the file alembic.ini contains the configuration needed to run alembic commands, and the directory alembic/ contains the Python code that executes migrations.

The directory looks like this.

```
alembic
  env.py
  script.py.mako
  versions
    15ac9509fc68_init.py
    faac8092c123_enable_security_pragmas.py
alembic.ini
```

The subdirectory versions/ individual migrations that are generated by alembic. In the example above, there are two migrations. alembic orders these migrations based off of values in the Python files, not off any sort of lexicographic ordering. The file faac8092c123_enable_security_pragmas.py has a module-level documentation string that specifies that it comes after 15ac9509fc68_init.py as well as variables used by alembic that specify the ordering of migrations.

### 60.2 Deployment

Database migrations are automatically applied to production instances via the command alembic upgrade head in the postinst script in the securedrop-app-code Debian package. You do not need to worry about when or how these migrations are applied.
60.3 Developer Workflow

60.3.1 Updating the Models

When you want to modify the database schema, you need to add adjust the models in the file models.py. All indices, constraints, or other metadata about the scheme needs to be in this file. The development server creates tables directly from the subclasses of db.Model so that they are available for manual and automated testing.

60.3.2 Creating Migrations

Once you are satisfied with your new model, alembic can auto-generate migrations using SQLAlchemy metadata and comparing it to the schema of an up-to-date SQLite database. To generate a new migration use the following steps.

```
cd securedrop/
./bin/dev-shell
source bin/dev-deps
maybe_create_config_py
./bin/new-migration 'my migration message'
```

This will output a new migration into alembic/versions/. You will need to verify that this migration produced the desired output. While still in the dev-shell, you can run the following command to see an output of the SQL that will be generated.

```
alembic upgrade head --sql
```

60.3.3 Unit Testing Migrations

The test suite already comes with a test runner (test_alembic.py) that runs a series of checks to ensure migration’s upgrade and downgrade commands are idempotent and don’t break the database. The test runner uses dynamic module import to iterate through all the migrations. You will need to create a python module in the tests/migrations/ directory. You module MUST be named migration_<revision identifier>.py. For example, if your revision is named 15ac9509fc68_init.py, your test module will be named migration_15ac9509fc68.py. Example modules for the first two revisions are shown below.

```
tests/migrations/
|-- __init__.py
|  |-- migration_15ac9509fc68.py
|  |-- migration_faac8092c123.py
```

Your module MUST contain the following classes with the following attributes.

```
class UpgradeTester:
  
  def __init__(self, config):
      '''This function MUST accept an argument named 'config'.
      You will likely want to save a reference to the config in your
      class so you can access the database later.'''
      self.config = config

  def load_data(self):
      '''This function loads data into the database and filesystem. It is
      executed before the upgrade.'''
```

(continues on next page)
Your migration test needs to load data that covers all edge cases such as potentially broken foreign keys or columns with unexpected content.

Additionally, your test MUST NOT import anything from the models module as this will not accurately test your migration, and it will likely break during future code changes. In fact, you should use as few dependencies as possible in your test including other securedrop code as well as external packages. This may be a rather annoying requirement, but it will make the tests more robust against future code changes.

### 60.3.4 Release Testing Migrations

In order to ensure that migrations between from the previous to current version of SecureDrop apply cleanly in production-like instances, we have a helper script that is designed to load semi-randomized data into the database. You will need to modify the script qa_loader.py to include sample data. This sample data should intentionally include edge cases that might behave strangely such as data whose nullability is only enforced by the application or missing files.

During QA, the release manager should follow these steps to test the migrations.

1. Checkout the previous SecureDrop release
2. Build Debian packages locally
3. Provision staging VMs
4. `vagrant ssh app-staging`
5. `sudo -u www-data bash`
6. `cd /var/www/securedrop && ./qa_loader.py`
7. Checkout the release candidate
8. Re-provision the staging VMs
9. Check that nothing went horribly wrong
SecureDrop is translated into a number of languages. We use a web-based collaborative translation platform called Weblate to make it easier. Under the hood, all translation is done using GNU gettext.

With gettext, text to be translated is specially marked in source code. A Python example:

```python
if not (msg or fh):
    flash(gettext("You must enter a message or choose a file to submit."), "error")
    return redirect(url_for('main.lookup'))
```

In this code, the string You must enter a message or choose a file to submit. can be automatically extracted for translation. The gettext function to which it is passed is used as a marker by pybabel or similar tools to collect the strings to be translated and store them into a .pot file at securedrop/translations/messages.pot. For instance:

```plaintext
#: source_app/main.py:111
msgid "You must enter a message or choose a file to submit."
msgstr "Vous devez saisir un message ou sélectionner un fichier à envoyer."
```

The .pot file serves as a template for all the language-specific .po files, which are where Weblate stores the contributed translations. For each language to be translated, a directory is created, such as securedrop/translations/fr_FR, and populated with a .po file derived from the template. For instance, securedrop/translations/fr_FR/LC_MESSAGES/messages.po is almost identical to securedrop/translations/messages.pot except for the msgstr fields, which will contain the French translations, e.g.:

```plaintext
#: source_app/main.py:111
msgid "You must enter a message or choose a file to submit."
msgstr "Vous devez saisir un message ou sélectionner un fichier à envoyer."
```

There’s one last type of file in the gettext system, a machine-readable version of the .po translations called a .mo file. Applications use these to get translations at runtime. The .po files are compiled to .mo files when the SecureDrop package is installed.

The Weblate web application is used to translate strings and relies on gettext behind the scenes. It owns the .pot and .po files. When preparing a SecureDrop release, a pull request is created to pull in all the translations that have been updated in Weblate.
The desktop icons installed on SecureDrop workstations are also translated. The icon templates are in the `install_files/ansible-base/roles/tails-config/templates` directory. Their labels are collected in the `desktop.pot` file and translated in the corresponding `.po` files in the same directory (fr. po, de.po etc.). All translations are merged from the `*.j2.in` files into the corresponding `*.j2` file and committed to the SecureDrop repository. They are then installed when configuring Tails with the `tasks/create_desktop_shortcuts.yml` tasks.

### 61.1 i18n\_tool.py

Most of the work in managing translations within the SecureDrop code base is supported by `securedrop/i18n_tool.py`. It provides convenient wrappers around `pybabel` and `gettext`, and is used to update strings to be translated; pull translations from Weblate; to compile translations before running tests and while packaging SecureDrop.

### 61.2 Development tasks

#### 61.2.1 Add a new language

A user with weblate admin rights must visit the Weblate translation creation page and the Weblate desktop translation creation page to add a new language. SecureDrop only supports a subset of all the languages being worked on in Weblate: some of them are partially translated or not fully reviewed. The list of fully supported languages is hard-coded in the `i18n_tool.py` file, in the `SUPPORTED_LANGUAGES` variable. When a new language is completely translated and reviewed, the `i18n_tool.py` file must be manually edited to add this new language to the `SUPPORTED_LANGUAGES` variable.

#### 61.2.2 Update strings to be translated

After strings are modified in the SecureDrop source code, templates or desktop labels, the `securedrop/translations/messages.pot` files must also be updated. Individual developers should not do this whenever changing strings in the code; the translations are updated in bulk when it’s time to update the Weblate fork. Translations can be updated with the following command:

```
$ make translate
```

This wraps `i18n_tool.py translate-messages` and `i18n_tool.py translate-desktop`. These commands update the `.pot` files for the SecureDrop server code and the desktop icons, as well as the `.po` files for each language.

**Note:** The new source strings will only be visible to translators in Weblate after the `develop` branch is merged into the Weblate fork.

#### 61.2.3 Merge develop into the Weblate fork

Weblate works on a long standing fork of the SecureDrop git repository and is exclusively responsible for the content of the `*.pot` and `*.po` files. The content of the `develop` branch must be merged into the `i18n` branch to make updated source strings available to Weblate.
Translation must be suspended in Weblate, and any uncommitted changes committed and pushed, to avoid conflicts:

- Go to the Weblate repository page for SecureDrop.

  - Click **Commit**.
  - Click **Push**.
  - And finally, click **Lock**.

The develop branch can now be merged into i18n:

```bash
$ git clone https://github.com/freedomofpress/securedrop
$ cd securedrop
$ git remote add i18n git@github.com:freedomofpress/securedrop-i18n.git
$ git fetch i18n
$ git checkout -b i18n i18n/i18n
$ git merge origin/develop
$ make -C securedrop translate
```

The `translate` Makefile target uses the `i18n_tool.py` command to keep the `*.pot` and `*.po` files in sync with
the SecureDrop source code. After running \texttt{make -C securedrop translate}, carefully review the output of \texttt{git diff}. Check \texttt{messages.pot} first for updated strings, looking for formatting problems. Then review the \texttt{messages.po} of one existing translation, with a focus on fuzzy translations. There is no need to review other translations because they are processed in the same way. When you are satisfied with the result, it can be merged with:

\begin{verbatim}
$ git commit -a -m 'l10n: sync with upstream origin/develop'
$ git push i18n i18n
\end{verbatim}

- Go to the Weblate commit page for SecureDrop and verify the commit hash matches the last commit of the i18n branch. This must happen instantly after the branch is pushed because Weblate is notified via a webhook. If it is different, ask for help.
- Click Unlock.

Weblate pushes the translations done via the web interface to the develop branch in a fork of the SecureDrop git repository. These commits must be manually cherry-picked and proposed as pull requests for the SecureDrop git repository.

61.2.4 Merge translations back to develop

Weblate automatically pushes the translations done via the web interface as a series of commits to the i18n branch in the Weblate SecureDrop branch, which is a fork of the develop branch of the SecureDrop git repository. These
translations need to be submitted back to the develop branch via pull requests. When you create a branch for this, begin its name with i18n-, as that prefix triggers special CI tests for translations.

To fetch the latest translations from the i18n branch into your working copy of the SecureDrop repository, run these commands in your repo root:

```
$ git checkout -b i18n-merge origin/develop
$ securedrop/bin/dev-shell ./i18n_tool.py --verbose update-from-weblate
$ securedrop/bin/dev-shell ./i18n_tool.py --verbose update-docs
```

You now have the latest translations on your i18n-merge branch.

**Note:** It is very important to check that each translated string looks like a plausible translation, with no markup. Even if the reviewer does not understand the language, if a translated string looks strange, someone other than the reviewer must be consulted to verify it means something. It is extremely unlikely that a contributor will manipulate a translated string to introduce a vulnerability in SecureDrop, but any suspicious translation should be investigated.

To check the new translations, you’ll need to compile them and verify them by running our automated tests and, ideally, by checking them in the SecureDrop source and journalist interfaces.

### Compile translations

At runtime, gettext needs a compiled file for each language (the .mo files). Before you can check the translations in the SecureDrop web interfaces, these need to be created:

```
$ securedrop/bin/dev-shell ./i18n_tool.py --verbose translate-messages --compile
```

For the desktop icons of the source and journalist interfaces, compilation updates their template files with all the translations collected from the .po files.

This can be done by running the following command:

```
$ securedrop/bin/dev-shell ./i18n_tool.py --verbose translate-desktop --compile
```

### Verify translations

#### SecureDrop web interfaces

After a translation is compiled, the web page in which it appears can be verified visually by starting the SecureDrop development servers and navigating via http://localhost:8080 for the source interface or http://localhost:8081 for the journalist interface. You can start the development servers with:

```
$ make -C securedrop dev
```

The translations can be checked automatically by running the SecureDrop page layout tests:

```
$ export PAGE_LAYOUT_LOCALES="en_US,fr_FR"  # may be set to any supported languages
$ make -C securedrop test TESTFILES=tests/pageslayout
[...]
tests/pageslayout/test_journalist.py::TestJournalistLayout::test_account_edit_hotp_secret[en_US] PASSED
tests/pageslayout/test_journalist.py::TestJournalistLayout::test_account_edit_hotp_secret[fr_FR] PASSED
[...]```
Note: if unset, PAGE_LAYOUT_LOCALES defaults to en_US (US English) and ar (Arabic).

After running the tests, screenshots for each locale are available in securedrop/tests/pageslayout/screenshots/<locale>, e.g. securedrop/tests/pageslayout/screenshots/fr_FR. Screenshots filenames can be found in the tests that created them, in securedrop/tests/pageslayout/test_journalist.py or securedrop/tests/pageslayout/test_source.py.

Desktop icons

The translated templates for the desktop icons are:

- install_files/ansible-base/roles/tails-config/templates/desktop-journalist-icon.j2
- install_files/ansible-base/roles/tails-config/templates/desktop-source-icon.j2

Check that each of them contains a Name line for each of SecureDrop’s supported locales.

Push your branch and create a pull request

After you’ve checked the translations, you’re ready to push your i18n-merge branch and create a pull request to get the translations merged to the SecureDrop develop branch.

Note: If there have been multiple commits per language, as can happen if source strings need to be translated again after being changed to correct critical errors, or to incorporate suggestions from the source string feedback period, they should be combined via an interactive rebase. Reorder the commits to group them by language, then squash the commits for each language into one. The goal is to end up with one commit per supported language on the merge branch.

When you’re happy with the state of language commits on your merge branch:

```
$ git commit -m "i18n: compile desktop icons' translations" # if needed
$ git push i18n-merge
```

Note: The CI job translation-tests will automatically run the above page layout tests in all supported languages on branches named with the prefix i18n-. If you’ve followed that naming convention, the translation tests should soon be run on your pull request.

If you have an abundance of time, you can run all the translation tests locally with:

```
$ make translation-test
```

And at long last, you’re done. Go to https://github.com/freedomofpress/securedrop and propose a pull request.

Note: Unlike the SecureDrop application translations, the desktop icon translations are compiled and merged into the repository. They need to be available in their translated form when securedrop-admin tailsconfig is run, because the development environment is not available.
61.3 Release Management

61.3.1 Two weeks before the release: string freeze

When features for a new SecureDrop release are frozen, the localization manager for the release will:

• **Merge develop into the Weblate fork.**
• **Post an announcement to the translation section of the forum (see an example).**
• **Add a prominent Weblate whiteboard announcement that reads The X.Y.Z deadline is Month day, year at midnight US/Pacific time. String freeze is in effect: no source strings will be modified before the release.**
• **Remind all developers about the string freeze, in the chat room.**
• **Create a pull request for every source string suggestion coming from translators.**
• **Update the i18n timeline and Weblate whiteboard.**

61.3.2 Release day

• **Merge translations back to develop.**
• **Update the screenshots.**
• **Remove the prominent Weblate whiteboard announcement.**
• **Provide translator credits to add to the SecureDrop release announcement.**
• **Update the i18n timeline and Weblate whiteboard.**

61.3.3 Translator credits

Correct acknowledgment of translators’ contributions is important, so `i18n_tool.py` makes it easy to list the translators who have helped since the last merge of Weblate translations, with `i18n_tool.py list-translators`. A list of everyone who has ever contributed translations to SecureDrop can be obtained with `i18n_tool.py list-translators --all`. There are Makefile targets for these, `list-translators` and `list-all-translators`, e.g:

```
$ make list-all-translators
ar:
  A. Nonymous
  Ahmad Gharbeia
  Ahmed Essam
  Ali Boshanab
  [...]`
```

61.4 Weblate administration

**Note:** The privilege escalation workflow is different for *code maintainers* and *translation maintainers*.

A translation admin has special permissions on Weblate and the repositories. When someone is willing to become an admin, a thread is started in the translation section of the forum. If there is consensus after a week, the permissions of the new admin are elevated. If there is not yet consensus, a public vote is organized among the current admins.
The privileges of an admin who has not been active for six months or more are revoked, but they can apply again at any time.

The community of SecureDrop translators works very closely with the SecureDrop developers and some of them participate in both groups. However, the translator community has a different set of rules and permissions, and therefore independent policies from SecureDrop itself.

61.4.1 Admin permissions

The full set of admin permissions can be granted at:

- https://weblate.securedrop.org/admin/weblate_auth/user/ (grant staff and superuser status)
- https://forum.securedrop.org/admin/users/list/active (click on the user and Grant Moderation)
- https://github.com/freedomofpress/securedrop-i18n (make sure that the user has commit access)

61.4.2 Granting reviewer privileges in Weblate

- Visit https://weblate.securedrop.org/admin/weblate_auth/user/.
- Click on the user name.
- In the Groups block:
  - Select Localizationlab in the Available groups list and click on the right arrow to move it to the Chosen groups list.
  - Select Users in the Chosen groups list and click on the left arrow to remove it.

61.4.3 Update the Weblate full text index

Weblate’s full-text index can occasionally get out of sync. When this happens, Weblate’s search may fail to find a word that you know exists in the source strings. You can rebuild the index with:

```
$ ssh debian@weblate.securedrop.org
$ cd /app/weblate
$ sudo docker-compose run weblate rebuild_index --all --clean
```

Note that the new index may not be used right away. Some workers may still have the old index open. If the index is holding up translators with a release looming, the server can be rebooted.
Translating SecureDrop

62.1 Putting SecureDrop in context for translation

62.1.1 What is SecureDrop?

SecureDrop is a system that lets people share sensitive information with investigative journalists anonymously and securely. It’s designed to protect its users with strong cryptography and network communications that hide locations and activity. For more information:

- Learn about what makes SecureDrop unique.
- Watch The Globe and Mail guide to using SecureDrop.
- Read the Localization Lab “Ask Me Anything” on SecureDrop.

62.1.2 Who Uses SecureDrop?

There are two kinds of SecureDrop users: Sources and Journalists. A source is an individual who wants to communicate securely and anonymously with a journalist. Sources are not expected to have any technical background. Journalists using SecureDrop have usually received proper training and understand the basic workflow of SecureDrop.

62.1.3 How is SecureDrop translated?

SecureDrop is translated using the Weblate platform. Read on to find out how to use Weblate.

62.2 Using Weblate

62.2.1 Registration

The first step is registering an account on our Weblate server by visiting the Weblate registration page.
You can choose to register with your email address, or by linking a GitHub account. If you already have a GitHub account to which you’re usually logged in, that can be easier, but it’s not required.

Register with email

If you want to just register with your email address, you’ll use the Register using email form on the left side of the registration page.

Fill the form and click Register, then check your email for a message from weblate@securedrop.org with the subject [Weblate] Your registration on Weblate. That message will contain a confirmation link. Clicking that link will complete your registration and redirect you to the Weblate dashboard.

Register with GitHub

To register with your GitHub account, click on the GitHub icon on the right, under Third party registration. This should take you to a page on https://github.com asking you to authorize the SecureDrop Weblate server to ask GitHub to authenticate you. (If you’re not already logged in to GitHub, you’ll be asked to do that first.) The authorization request looks like this:
Click the green Authorize freedomofpress button, and you will be redirected to the Weblate dashboard.

### 62.2.2 Choosing your preferred languages

You can contribute to any language, but Weblate has some conveniences to make it easier to work with your preferred languages. You can manage those from the Weblate dashboard:
Click the **Manage your languages** button.

In each selection box, simply click on the languages you would like to translate. When you’re done, click the **Save** button at the bottom of the page.

### 62.2.3 The SecureDrop translation project

Webate supports multiple projects. Our instance only contains one, SecureDrop, which has two translation components: the main SecureDrop web application (labeled “SecureDrop”), and the translations for the desktop icons of the admin and journalist workstations used by news organizations (labeled “desktop”).

On the dashboard you will find links to both sets of translations. You can also drill down to them via the **Projects** menu in the top navigation bar.
If you click on a component, you’ll arrive at the list of languages into which it’s translated:

To start translating a language, click the **Translate** button at the far right of its row.

### 62.2.4 Translating a phrase

Each translatable string is shown in the text area labeled **Source**. You can translate it right below in the **Translation** text area. When you are done, click **Save** and the next untranslated string will appear.
On the right of the page you can find helpful information about the source string. If the language glossary contains terms from the source string, their suggested translations will appear in the Glossary sidebar. In the Source information sidebar, you’ll find contextual information about the source string, like its location in our source code. If a screenshot showing the string in context in the SecureDrop user interface is available, that might be linked here as well.

## 62.3 Placeholders

Source strings may contain placeholder text in curly braces, for example \{count\}. These represent variable content (like a username, as in the example below), and must be left unmodified, but they can be moved around in a string. For instance:

```
Edit user {user}
```

might be displayed to the user as:

```
Edit user Jean-Claude
```

The French translated string should look like:

```
Modifier l'utilisateur \{user\}
```

And it would be incorrect to translate the placeholder like so:

```
Modifier l'utilisateur \{utilisateur\}
```

### 62.3.1 HTML content

Some source strings represent HTML that will be presented in the SecureDrop web interface. It can be hard to tell what to translate, since HTML is source code and changing the wrong thing can break the page layout.

One thing you should always translate are alt attributes of image elements. Image elements \(<img>\) in HTML place a picture on the page. People with visual impairments rely on a special note on the image element – the alt attribute
– to describe the image, so it’s helpful to translate those. Here’s an example that contains an image with both an alt attribute and a placeholder:

```html
<img src="{icon}" alt="shield icon">
```

As explained above, the placeholder `{icon}` in the `src` attribute of the `<img>` element should not be translated. The `alt` attribute text ("shield icon") should be. The correctly translated HTML in Portuguese would be:

```html
<img src="{icon}" alt="ícone do escudo">
```

### 62.4 Reviews

Translated strings must be approved by a reviewer before being accepted into SecureDrop. This is to make sure the source or journalist will not be confused by an incorrect translation.

Anyone can contribute translations, just like anyone can edit Wikipedia. However the right to review translations is only extended to trusted translators. You can ask to become a reviewer for a language by posting a message in the translation category of the SecureDrop forum.

A reviewer sees a Review state section below the source string, including the Waiting for review and Approved radio buttons. All source strings are initially set to Waiting for review.

When the translation is deemed correct, the reviewer should change it to Approved.

Once a translation has been approved, only reviewers can modify it. Translators can still suggest modifications or make comments if they notice something wrong.

### 62.5 Glossaries

Translating SecureDrop requires understanding some complicated security terminology, involving cryptography, networking, and threat models. If you are unfamiliar with terms you encounter in source strings, consult one of the following glossaries, or ask in the translation category of the SecureDrop forum.

- The SecureDrop glossary explains terms specific to SecureDrop.
• The EFF Surveillance Self-Defense glossary explains many general security concepts.

62.5.1 Weblate Glossary

Weblate also contains an internal glossary for each language, to which we can add suggested translations. If a source string contains terms from this glossary, the glossary entries will be displayed in a box on the right side of the translation page.

If you find that a source string contains terms from the SecureDrop glossary or the EFF Surveillance Self-Defense glossary, but the glossary sidebar says No related strings found in the glossary, we’d really appreciate it if you could add those terms to the glossary of the language you’re working with. Simply fill out the Source and Translation fields in the sidebar under Add new word to glossary, then click the Add button.

You can see the full glossary for a language by clicking the edit icon at the top right of the glossary sidebar. The glossary page contains all the terms that have been translated for the language:

62.6 Getting Help

Should you need help, you can do one of the following:

• Post a message in the translation category of the SecureDrop forum
• Chat in the SecureDrop instant messaging channel
• Read the Weblate documentation
62.7 Collaboration Guidelines

62.7.1 Adding new languages

We love seeing SecureDrop translated into new languages. Just ask us to add yours by posting in the translation category of the SecureDrop forum.

62.7.2 Suggesting changes to source strings

If you notice errors in our source strings, or catch us using English idioms that are hard to translate, please add comments letting us know. At the beginning of every translation cycle in our release schedule, we have a few days for incorporating your feedback, and very much appreciate it.

62.7.3 Changing an existing translation

If you think a translation can be improved, please don’t run roughshod over another translator’s work. Make a suggestion or comment first, to allow discussion before saving your changes.

Exceptions to this policy would be:

- Obvious errors in spelling, grammar, or punctuation.
- A string in our interface that is supposed to match another project. For example, we include instructions for adjusting Tor Browser settings, so if our wording is out of date, it has to be corrected to reduce confusion for people using SecureDrop.

In those cases, please feel free to correct the existing translation.
Documentation Guidelines

SecureDrop’s documentation is written in ReStructuredText (ReST), and is built by and hosted on Read the Docs (RTD). The documentation files are stored in the primary SecureDrop git repository under the docs/ directory.

To get started editing the docs:

1. Clone the SecureDrop repository:

   ```
   git clone https://github.com/freedomofpress/securedrop.git
   ```

2. Install the dependencies:

   ```
   pip install -r securedrop/requirements/develop-requirements.txt
   ```

3. Build the docs for viewing in your web browser:

   ```
   make docs
   ```

You can then browse the documentation at http://127.0.0.1:8000/. As you make changes, the documentation pages will automatically rebuild in the browser window, so you don’t need to refresh the page manually.

### 63.1 Testing Documentation Changes

You can check for formatting violations by running the linting option:

```
make docs-lint
```

The `make docs` command will display warnings, but will still build the documentation if formatting mistakes are found. Using `make docs-lint` will convert any warnings to errors, causing the build to fail. The CI tests will automatically perform linting via the same command.

To test the documentation for broken links, run the following command from a reliable internet connection:
The CI tests by default create staging servers to test the application code. If your PR only makes documentation changes, you should prefix the branch name with docs- to skip the staging run. Project maintainers will still need to approve the PR prior to merge, and the linting checks will also still run.

63.2 Updating Screenshots

The user guides for SecureDrop contain screenshots of the web applications. To update these screenshots automatically you can run:

```
make -C securedrop update-user-guides
```

This will generate screenshots for each page in the web application and copy them to the folder under docs/images/manual/screenshots where they will replace the existing screenshots. Stage for commit any screenshots you wish to update. If you wish to update all screenshots, simply stage for commit all changed files in that directory.

63.3 Integration with Read the Docs

Note: SecureDrop maintains two versions of documentation: stable and latest. stable is the default used by our Read the Docs site, and is built from our latest signed git tag. latest is built from the head of the develop git branch. In almost all cases involving development work, you’ll want to make sure you have the latest version selected by using the menu in the bottom left corner of the Read the Docs site.

Our documentation is built and hosted by Read the Docs and is available at https://docs.securedrop.org. We use a webhook to rebuild the documentation automatically when commits get pushed to the branch.

63.4 Style Guide

63.4.1 Line Wrapping

Lines in the plain-text documentation files should wrap at 80 characters. (Some exceptions: complex code blocks showing example commands, or long URLs.)

63.4.2 Glossary

Text taken directly from a user interface is in bold face.

“All you’re sure you have the right drive, click **Format Drive**.”

SecureDrop-specific glossary is in italics.

“To get started, you’ll need two Tails drives: one for the Admin Workstation and one for the Secure Viewing Station.”

When referring to virtual machines in the development environment, use lowercase for the name:

app-staging VM
63.4.3 Code Blocks

Ensure that example commands in codeblocks are easy to copy and paste. Do not prepend the $ shell prompt indicator to example commands:

```
echo hello
```

In the context of a terminal session with both typed commands and printed output text, use $ before the typed commands:

```
$ echo hello
hello
$ echo sunshine
sunshine
```

63.4.4 File Paths

Cloning the SecureDrop git repository creates a directory called `securedrop`. This `securedrop` directory also contains a `securedrop` subdirectory for app code.

```
.  └── securedrop
   │   └── securedrop
   ... ...
```

To avoid confusion, paths to files anywhere inside the SecureDrop git repository should be written as `.some_dir/file`, where . is the top level directory of the SecureDrop repo.

Use absolute paths when referring to files outside the SecureDrop repository: `/usr/local/bin/tor-browser`.

63.4.5 Usage and Style

To avoid confusion, lists should include the so-called “Oxford comma”:

“You will need an email address, a public GPG key for that address, and the fingerprint for that key.”

Capitalize all section headings in title case:

```
Before You Begin
-----------------
Read the Docs
--------------
```

not

```
Before you begin
-----------------
Read the docs
--------------
```
The SecureDrop project ships both application code for running on servers hosted on-site at news organizations, as well as configuration scripts for provisioning the servers to accept updates to the application code, and to harden the system state. Therefore testing for the project includes Application Tests for validating that the app code behaves as expected, and Configuration Tests to ensure that the servers are appropriately locked down, and able to accept updates to the app code.

In addition, the Continuous Integration automatically runs the above Application and Configuration tests against cloud hosts, to aid in PR review.
Testing: Application Tests

The application test suite uses:

- Pytest
- Selenium

The application tests consist of unit tests for the Python application code and functional tests that verify the functionality of the application code from the perspective of the user through a web browser.

The functional tests use an outdated version of Firefox chosen specifically for compatibility with Selenium 2, and a rough approximation of the most recent Tor Browser.

**Note:** We’re working on running the Selenium tests in Tor Browser. See GitHub #1629 for more info.

### 65.1 Installation

The application tests are installed automatically in the development and app-staging VMs, based on the contents of `securedrop/requirements/test-requirements.txt`. If you wish to change the dependencies, see *Upgrading or Adding Python Dependencies*.

### 65.2 Running the Application Tests

The tests can be run inside the development VM:

```
make -C securedrop test
```

Or the app-staging VM:
vagrant ssh app-staging
sudo bash
cd /var/www/securedrop
pytest -v tests
chown -R www-data /var/lib/securedrop /var/www/securedrop

**Warning:** The `chown` is necessary because running the tests as root will change ownership of some files, creating problems with the source and journalist interfaces.

For explanation of the difference between these machines, see *Virtual Environments: Servers*.

If you just want to run the functional tests, you can use:

```bash
securedrop/bin/dev-shell bin/run-test -v tests/functional
```

Similarly, if you want to run a single test, you can specify it through the file, class, and test name:

```bash
securedrop/bin/dev-shell bin/run-test \
  tests/test_journalist.py::TestJournalistApp::test_invalid_credentials
```

The `gnupg` library can be quite verbose in its output. The default log level applied to this package is `ERROR` but this can be controlled via the `GNUPG_LOG_LEVEL` environment variable. It can have values such as `INFO` or `DEBUG` if some particular test case or test run needs greater verbosity.

### 65.2.1 Page Layout Tests

You can check the rendering of the layout of each page in each translated language using the page layout tests. These will generate screenshots of each page and can be used for example to update the SecureDrop user guides when modifications are made to the UI.

You can run all tests, including the page layout tests with the `--page-layout` option:

```bash
securedrop/bin/dev-shell bin/run-test --page-layout tests
```

### 65.3 Updating the Application Tests

Unit tests are stored in the `securedrop/tests/` directory and functional tests are stored in the functional test directory:

```
supported_drop/tests/
  functional
  |   test_admin_interface.py
  |   test_submit_and_retrieve_file.py
  |   submission_not_in_memory.py
  ... 
  utils
  |   db_helper.py
  |   env.py
  |   asynchronous.py
  test_journalist.py
  test_source.py
```

(continues on next page)
...\ntest_store.py

securedrop/tests/utils contains helper functions for writing tests. If you want to add a test, you should see if there is an existing file appropriate for the kind of test, e.g. a new unit testing manage.py should go in test_manage.py.
Testing: Configuration Tests

Testinfra tests verify the end state of the staging VMs. Any changes to the Ansible configuration should have a corresponding specetest.

66.1 Installation

```
pip install -r securedrop/requirements/develop-requirements.txt
```

66.2 Running the Config Tests

In order to run the tests, first create and provision the VM you intend to test.

For the staging VMs:

```
make build-debs
make staging
```

The VMs will be set up using either the libvirt or virtualbox Vagrant VM provider, depending on your system settings. You’ll need to use the appropriate commands below based on your choice of provider.

Then, to run the tests:

66.2.1 libvirt:

```
molecule verify -s libvirt-staging
```
66.2.2 virtualbox:

```
molecule verify -s virtualbox-staging
```

**Tip:** To run only a single test, set `PYTEST_ADDOPTS="-k name_of_test"` in your environment.

Test failure against any host will generate a report with informative output about the specific test that triggered the error. Molecule will also exit with a non-zero status code.

**Note:** To build and test the VMs with one command, use the Molecule test action:

```
molecule test -s libvirt-staging --destroy=never, or molecule test -s virtualbox-staging --destroy=never.
```

### 66.3 Updating the Config Tests

Changes to the Ansible config should result in failing config tests, but only if an existing task was modified. If you add a new task, make sure to add a corresponding spectest to validate that state after a new provisioning run. Tests import variables from separate YAML files than the Ansible playbooks:

```
molecule/testinfra/staging/vars/
├── app-prod.yml
├── app-staging.yml
├── mon-prod.yml
├── mon-staging.yml
├── staging.yml
```

Any variable changes in the Ansible config should have a corresponding entry in these vars files. These vars are dynamically loaded for each host via the `molecule/testinfra/staging/conftest.py` file. Make sure to add your tests to the relevant location for the host you plan to test:

```
molecule/testinfra/staging/app/
├── apache
│   ├── test_apache_journalist_interface.py
│   ├── test_apache_service.py
│   ├── test_apache_source_interface.py
│   └── test_apache_system_config.py
├── test_apparmor.py
├── test_appenv.py
├── test_network.py
└── test_ossec.py
```

In the example above, to add a new test for the `app-staging` host, add a new file to the `testinfra/staging/app` directory.

**Tip:** Read *Updating OSSEC Rules* to learn how to write tests for the OSSEC rules.
66.4 Config Test Layout

With some exceptions, the config tests are broken up according to platform definitions in the Molecule configuration:

```
molecule/testinfra/staging
  ├── app
  │    └── app-code
  ├── common
  │    └── mon
  └── ossec
      └── vars
```

Ideally the config tests would be broken up according to roles, mirroring the Ansible configuration. Prior to the reorganization of the Ansible layout, the tests are rather tightly coupled to hosts. The layout of config tests is therefore subject to change.

66.5 Config Testing Strategy

The config tests currently emphasize testing implementation rather than functionality. This was a temporary measure to increase the testing baseline for validating the Ansible provisioning flow, which aided in migrating to a current version of Ansible (v2+). Now that the Ansible version is current, the config tests can be improved to validate behavior, such as confirming ports are blocked via external network calls, rather than simply checking that the iptables rules are formatted as expected.
The SecureDrop project uses CircleCI for running automated test suites on code changes. The relevant files for configuring the CI tests are the Makefile in the main repo, the configuration file at .circleci/config.yml, and the scripts in devops/. You may want to consult the CircleCI Configuration Reference to interpret the configuration file. Review the workflows section of the configuration file to understand which jobs are run by CircleCI.

The files under devops/ are used to create a libvirt-compatible environment on GCE. The GCE host is used as the Ansible controller, mimicking a developer’s laptop, to provision the machines and run the tests against them.

**Note:** We skip unnecessary jobs, such as the staging run, for pull requests that only affect the documentation; to do so, we check whether the branch name begins with docs-. These checks are enforced in different parts of the configuration, mainly within the Makefile.

**Warning:** In CI, we rebase branches in PRs on HEAD of the target branch. This rebase does not occur for branches that are not in PRs. When a branch is pushed to the shared freedomofpress remote, CI will run, a rebase will not occur, and since opening a PR does not trigger a re-build, the CI build results are not shown rebased on the latest of the target branch. This is important to maintain awareness of if your branch is behind the target branch. Once your branch is in a PR, you can rebuild, push an additional commit, or manually rebase your branch to update the CI results.

### 67.1 Running the CI Staging Environment

The staging environment tests will run automatically in CircleCI, when changes are submitted by Freedom of the Press Foundation staff (i.e. members of the freedomofpress GitHub organization). The tests also perform basic linting and validation, like checking for formatting errors in the Sphinx documentation.
Tip: You will need a Google Cloud Platform account to proceed. See the Google Cloud Platform Getting Started Guide for detailed instructions.

In addition to a GCP account, you will need a working Docker installation in order to run the container that builds the deb packages.

You can verify that your Docker installation is working by running `docker run hello-world` and confirming you see “Hello from Docker” in the output as shown below:

```
$ docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.
...
```

### 67.1.1 Setup Environment Parameters

Source the setup script using the following command:

```
source ./devops/gce-nested/ci-env.sh
```

You will be prompted for the values of the required environment variables. There are some defaults set that you may want to change. You will need to export `GOOGLE_CREDENTIALS` with authentication details for your GCP account, which is outside the scope of this guide.

### 67.1.2 Use Makefile to Provision Hosts

Run `make help` to see the full list of CI commands in the Makefile:

```
$ make help
Makefile for developing and testing SecureDrop.
Subcommands:
  ci-go               Creates, provisions, tests, and destroys GCE host for
                   → testing staging environment.
  ci-go-xenial       Creates, provisions, tests, and destroys GCE host for
                   → testing staging environment under xenial.
  ci-lint            Runs linting in linting container.
  ci-teardown        Destroys GCE host for testing staging environment.
```

To run the tests locally:

```
make ci-go
```

You can use `./devops/gce-nested/ci-runner.sh` to provision the remote hosts while making changes, including rebuilding the Debian packages used in the Staging environment. See Virtual Environments: Servers for more information.
SecureDrop apt Repository

This document contains brief descriptions of the Debian packages hosted and maintained by Freedom of the Press Foundation in our apt repository (apt.freedom.press).

**linux-image-4.4.*-grsec**  This package contains the Linux kernel image, patched with grsecurity. Listed as a dependency of securedrop-grsec.

**ossec-agent**  Installs the OSSEC agent, repackaged for Ubuntu. Listed as a dependency of securedrop-ossec-agent.

**ossec-server**  Installs the OSSEC manager, repackaged for Ubuntu. Listed as a dependency of securedrop-ossec-server.

**securedrop-app-code**  Packages the SecureDrop application code, Python pip dependencies and AppArmor profiles.

**securedrop-ossec-agent**  Installs the SecureDrop-specific OSSEC configuration for the *Application Server*.

**securedrop-ossec-server**  Installs the SecureDrop-specific OSSEC configuration for the *Monitor Server*.

**securedrop-grsec**  SecureDrop grsecurity kernel metapackage, depending on the latest version of linux-image-3.14-*-grsec.

**securedrop-keyring**  Packages the public signing key for this apt repository. Allows for managed key rotation via automatic updates, as implemented in SecureDrop 0.3.10.

**Note:**  The SecureDrop install process configures a custom Linux kernel hardened with the grsecurity patch set. Only binary images are hosted in the apt repo. For source packages, see the Source Offer.
SecureDrop uses the OSSEC open source host-based intrusion detection system (IDS) for log analysis, file integrity checking, policy monitoring, rootkit detection and real-time alerting. Refer to our OSSEC guide to learn more about how SecureDrop admins set up and monitor OSSEC alerts.

### 69.1 Alerting Strategy

The goals of the OSSEC alerts in SecureDrop is to notify admins of:

1. Suspicious security events
2. Changes that require some kind of admin action
3. Other important notifications regarding system state.

If an alert is purely informational and there is no realistic action an admin is expected to take, you should think carefully before suggesting a rule for it. Each additional alert that admins must read and/or respond to takes time. Alerts that are unimportant or otherwise require no action can lead to alert fatigue and thus to critical alerts being ignored.

### 69.2 Using ossec-logtest

Development on the OSSEC rules should be done from the staging environment.

On mon-staging, there is a utility installed as part of OSSEC called ossec-logtest that you can use to test log events. In order to evaluate whether an alert will be produced, and if so, what rule triggered it and its level, you can simply pass the event to ossec-logtest:
This is the utility we use in automated tests of OSSEC.

## 69.3 Writing Automated Tests for OSSEC Rules

We strongly recommend before making changes to OSSEC rules to attempt to write a failing test which you then can make pass with a patch to the OSSEC rules:

1. Identify a log event you can use to trigger the alert.

```bash
Warning: Be sure to use only log events from test SecureDrop instances or those you have verified do not contain any sensitive data.
```

2. Write a Testinfra test to verify that the log event does or does not trigger an alert.

3. Apply your patch to the OSSEC rule on the relevant VM (likely app).

4. Restart the service via `sudo service ossec restart` on `mon`.

```bash
Note: Currently we only have automated tests for alerts triggered due to log events (for example not for syscheck, OSSEC's integrity checking process). If you have ideas for additional automated test coverage of alerts, please suggest them in ticket 2134 on GitHub.
```
69.4 Deployment

The OSSEC rules and associated configuration files are distributed via Debian packages maintained by Freedom of the Press Foundation. Any changes made to OSSEC configuration files will land on production SecureDrop monitoring servers as part of each SecureDrop release. This upgrade will occur automatically.

Note: The use of automatic upgrades for release deployment means that any changes made locally by admins to their OSSEC rules will not persist after a SecureDrop update.
Generating AppArmor Profiles for Tor and Apache

```bash
vagrant up /staging$
vagrant ssh app-staging
sudo su
cd /var/www/securedrop

Run tests, use the application web interface, restart services, reboot the VMs via `vagrant reload /staging/`. The goal is to create as much interaction with the system as possible, in order to establish an expected baseline of behavior. Then run:

```
aa-logprof
```

Follow the prompts on screen and save the new configuration. Then set the profile to complain mode:

```
aa-complain /etc/apparmor.d/<PROFILE_NAME>
```

Rinse and repeat, again running `aa-logprof` to update the profile. The AppArmor profiles are saved in `/etc/apparmor.d/`. There are two profiles:

- `/etc/apparmor.d/usr.sbin.tor`
- `/etc/apparmor.d/usr.sbin.apache2`

After running `aa-logprof` you will need to copy the modified profile back to your host machine to include them in the `securedrop-app-code` package.

```bash
ansible -i .vagrant/provisioners/ansible/inventory/vagrant_ansible_inventory app-prod
  -m fetch -a 'flat=yes dest=install_files/ansible-base/ src=/etc/apparmor.d/usr.sbin.apache2'
ansible -i .vagrant/provisioners/ansible/inventory/vagrant_ansible_inventory app-prod
  -m fetch -a 'flat=yes dest=install_files/ansible-base/ src=/etc/apparmor.d/usr.sbin.tor'
```

The AppArmor profiles are packaged with the `securedrop-app-code`. The `securedrop-app-code postinst` puts the AppArmor profiles in enforce mode on production and staging hosts.
When at a conference or traveling, it is possible to prepare a SecureDrop demo using portable hardware and adapted usage scenarios.

### 71.1 Hardware

- A laptop running the *staging virtual environment*
- A Tails compatible laptop with a physical radio kill switch (for instance a Lenovo T420)
- Four USB keys prepared for the staging environment running on the laptop
  - Transfer
  - Journalist
  - SVS
  - Admin

The Tails compatible laptop has the physical radio kill switch turned off to simulate a SVS and it is rebooted with the physical radio kill switch turned on to simulate the Admin or Journalist workstation.
The **Release Manager** is responsible for shepherding the release process to successful completion. This document describes their responsibilities. Some items must be done by people that have special privileges to do specific tasks (e.g. privileges to access the production apt server), but even if the **Release Manager** does not have those privileges, they should coordinate with the person that does to make sure the task is completed.

### 72.1 Pre-Release

1. Open a **Release SecureDrop 0.x.y** issue to track release-related activity. Keep this issue updated as you proceed through the release process for transparency.

2. Check if there is a new stable release of Tor that can be QAed and released as part of the SecureDrop release. If so, file an issue.

3. Check if a release candidate for the Tails release is prepared. If so, request people participating in QA to use the latest release candidate.

4. Prepare a changelog describing the changes in the release.

5. Ensure that a pre-release announcement is prepared and shared with the community for feedback. Once the announcement is ready, coordinate with other team members to send them to current administrators, post on the SecureDrop blog, and tweet out a link.

6. For a regular release for version 0.x.0, branch off develop:

   ```bash
   git checkout develop
   git checkout -b release/0.x
   ```

   **Warning:** For new branches, please ask a **freedomofpress** organization administrator to enable branch protection on the release branch. We want to require CI to be passing as well as at least one approving review prior to merging into the release branch.

7. Prepare each release candidate where `rcN` is the Nth release candidate using this script:
8. If you would like to sign the release commit, you will need to do so manually:
   a. Create a new signed commit and verify the signature:
      
      ```
      git reset HEAD~1
      git commit -aS
      git log --show-signature
      ```
   b. Ensure the new commit is signed, take note of the commit hash.
   c. Edit `0.x.y-rcN.tag` and replace the commit hash with the new (signed) commit hash.
   d. Delete the old tag and create a new one based on the tag file edited above:
      
      ```
      git tag -d 0.x.y-rcN
      git mktag < 0.x.y-rcN.tag > .git/refs/tags/0.x.y-rcN
      ```
9. Push the branch and tags.
10. Build Debian packages and place them on `apt-test.freedom.press`.
11. Write a test plan that focuses on the new functionality introduced in the release. Post for feedback and make changes based on suggestions from the community.
12. Encourage QA participants to QA the release on production VMs and hardware. They should post their QA reports in the release issue such that it is clear what was and what was not tested. It is the responsibility of the release manager to ensure that sufficient QA is done on the release candidate prior to final release.
13. Triage bugs as they are reported, if a bug is important to fix and does not receive attention, you should fix the bug yourself or find someone who agrees to work on a fix.
14. Backport release QA fixes merged into `develop` into the release branch using `git cherry-pick -x <commit>` to clearly indicate where the commit originated from.
15. At your discretion - for example when a significant fix is merged - prepare additional release candidates and have fresh Debian packages prepared for testing.
16. For a regular release, the string freeze will be declared by the translation administrator one week prior to the release. After this is done, ensure that no changes involving string changes are backported into the release branch.
17. Ensure that a draft of the release notes are prepared and shared with the community for feedback.

### 72.2 Release Process

1. If this is a regular release, work with the translation administrator responsible for this release cycle to review and merge the final translations and screenshots (if necessary) they prepare. Refer to the `i18n documentation` for more information about the i18n release process. Note that you must manually inspect each line in the diff to ensure no malicious content is introduced.
2. Prepare the final release commit and tag. Do not push the tag file.
3. Step through the signing ceremony for the tag file. If you do not have permissions to do so, coordinate with someone that does.
4. Once the tag is signed, append the detached signature to the unsigned tag:
5. Delete the original unsigned tag:

```
git tag -d 0.x.y
```

6. Make the signed tag:

```
git mktag < 0.x.y.tag > .git/refs/tags/0.x.y
```

7. Verify the signed tag:

```
git tag -v 0.x.y
```

8. Push the signed tag:

```
git push origin 0.x.y
```

9. Ensure there are no local changes (whether tracked, untracked or git ignored) prior to building the debs. If you did not freshly clone the repository, you can use `git clean`:

Dry run (it will list the files/folders that will be deleted):

```
git clean -ndfx
```

Actually delete the files:

```
git clean -dfx
```

10. Build Debian packages. People building Debian packages should verify and build off the signed tag.

11. Step through the signing ceremony for the Release file(s) (there may be multiple if Tor is also updated along with the SecureDrop release).


13. Coordinate with one or more team members to confirm a successful clean install in production VMs using the packages on `apt-test.freedom.press`.

14. Put signed Debian packages on `apt.freedom.press`. The release is now live.

15. Make sure that the default branch of documentation is being built off the tip of the release branch.

16. Create a release on GitHub with a brief summary of the changes in this release.

17. Make sure that release notes are written and posted on the SecureDrop blog.

18. Make sure that the release is announced from the SecureDrop Twitter account.

19. Make sure that members of the support portal are notified about the release.

## 72.3 Post-Release

After the release, carefully monitor the FPF support portal (or ask those that have access to monitor) and SecureDrop community support forum for any issues that users are having.

Finally, in a PR back to develop, cherry-pick the release commits (thus ensuring a consistent changelog in the future) and bump the version numbers in preparation for the next release (this is required for the upgrade testing scenario).
73.1 Get your Quay account squared away

The container that performs builds of Debian packages is version controlled in a docker repository at quay.io/freedomofpress/sd-docker-builder. There are tight restrictions over who can make edits here. If you have permissions to do so, you’ll need to make sure your local docker client has credentials to push.

- First login into your quay.io account via the web-portal at https://quay.io/
- Drill into your Account settings via the upper right drop-down (where your username is)
- Click Generate Encrypted Password
- From a command-line prompt type docker login quay.io with your username and credentials obtained from the previous step.
- Proceed with update instructions

73.2 Performing container updates

If one of the dependencies requires security updates, the build may fail at test_ensure_no_updates_avail. If you have access rights to push to quay.io, here is the process to build and push a new container:

```
$ cd molecule/builder/
$ # Build a new container
$ make build-container
```

Once the container is built, you can push the container to the registry.

```
$ make push-container
```

You can now test the container by going back to the SecureDrop repository root:
cd ../..
make build-debs

Assuming no errors here, commit the changes in molecule/builder/image_hash in a branch containing the prefix update-builder-.
Deploying SecureDrop staging instance on Qubes

This assumes you have an up-to-date Qubes installation on a compatible laptop with at least 16GB RAM and 60GB free disk space.

74.1 Overview

Follow the Qubes platform instructions in *Setting Up the Development Environment* to create an sd-dev Standalone VM. Once done, we’ll create three new Standalone (HVM) Qubes VMs for use with staging:

- **sd-staging-base**, a base VM for cloning reusable staging VMs
- **sd-app-base**, a base VM for the *SecureDrop Application Server*
- **sd-mon-base**, a base VM for the *SecureDrop Monitor Server*

While the development VM, sd-dev, is based on Debian 9, the other VMs will be based on Ubuntu Xenial.

74.2 Download Ubuntu Xenial server ISO

On sd-dev, download the Ubuntu Xenial server ISO, along with corresponding checksum and signature files. See the *hardware installation docs* for detailed instructions. If you opt for the command line instructions, omit the torify prepended to the curl command.

74.3 Create the base VM

We’re going to build a single, minimally configured Ubuntu VM. Once it’s bootable, we’ll clone it for the application and monitoring VMs.

In dom0, do the following:
The commands above will create a new StandaloneVM, expand the storage space and memory available to it, as well as disable the integrated kernel support. The SecureDrop install process will install a custom kernel.

### 74.4 Boot into installation media

In dom0:

```
qvm-start sd-staging-base --cdrom=sd-dev:/home/user/ubuntu-16.04.6-server-amd64.iso
```

You may need to edit the filepath above if you downloaded the ISO to a different location within the sd-dev VM. Choose **Install Ubuntu**. For the most part, the install process matches the *hardware install flow*, with a few exceptions:

- Server IP address: use value returned by `qvm-prefs sd-staging-base ip`, with /24 netmask suffix
- Gateway: use value returned by `qvm-prefs sd-staging-base visible_gateway`
- For DNS, use Qubes’s DNS servers: 10.139.1.1 and 10.139.1.2.
- Hostname: `sd-staging-base`
- Domain name should be left blank

Make sure to configure LVM and use **Virtual disk 1 (xvda 20.0GB Xen Virtual Block device)** when asked for a target partition during installation. It should be the default option.

You’ll be prompted to add a “regular” user for the VM: this is the user you’ll be using later to SSH into the VM. We’re using a standardized name/password pair: sdadmin/securedrop.

Once installation is done, let the machine shut down and then restart it with

```
qvm-start sd-staging-base
```

in dom0. You should get a login prompt.

### 74.5 Initial VM configuration

Before cloning this machine, we’ll update software to reduce provisioning time on the staging VMs. In the new sd-staging-base VM’s console, do:

```
sudo apt update
dsudo apt dist-upgrade -y
```

Before we continue, let’s allow your user to `sudo` without their password. Edit `/etc/sudoers` using `visudo` to make the sudo group line look like

```
%sudo   ALL=(ALL) NOPASSWD: ALL
```

Finally, update the machine’s Grub configuration to use a consistent Ethernet device name across kernel versions. Edit the file `/etc/default/grub`, changing the line:
GRUB_CMDLINE_LINUX=""  

to  

GRUB_CMDLINE_LINUX="net.ifnames=0 biosdevname=0"

When initial configuration is done, run qvm-shutdown sd-staging-base to shut it down.

### 74.6 Clone VMs

We're going configure the VMs to use specific IP addresses, which will make various routing issues easier later. We’ll also tag the VMs for management by the sd-dev VM. Doing so will require Qubes RPC policy changes, documented below. Run the following in dom0:

```
qvm-clone sd-staging-base sd-app-base
qvm-clone sd-staging-base sd-mon-base
qvm-prefs sd-app-base ip 10.137.0.50
qvm-prefs sd-mon-base ip 10.137.0.51
qvm-tags sd-app-base add created-by-sd-dev
qvm-tags sd-mon-base add created-by-sd-dev
```

Now start both new VMs:

```
qvm-start sd-app-base
qvm-start sd-mon-base
```

On the consoles which eventually appear, you should be able to log in with sdadmin/securedrop.

#### 74.6.1 Configure cloned VMs

We’ll need to fix each machine’s idea of its own IP. In the console for each machine, edit `/etc/network/interfaces` to update the address line with the machine’s IP.

Edit `/etc/hosts` on each host to include the hostname and IP for itself. Use `sd-app` and `sd-mon`, omitting the `-base` suffix, since the cloned VMs will not have the suffix.

Next, on each host edit `/etc/hostname` to reflect the machine’s name. Again, omit the `-base` suffix.

Halt each machine, then restart each from dom0. The prompt in each console should reflect the correct name of the VM. Confirm you have network access by running `host freedom.press`. It should show no errors.

#### 74.6.2 Inter-VM networking

We want to be able to SSH connections from sd-dev to these new standalone VMs. In order to do so, we have to adjust the firewall on `sys-firewall`.

**Tip:** See the official Qubes guide on configuring inter-VM networking for details.

Let’s get the IP address of sd-dev. On dom0:

```
qvm-prefs sd-dev ip
```
Get a shell on `sys-firewall`. Create or edit `/rw/config/qubes-firewall-user-script`, to include the following:

```bash
sd_dev="<sd-dev-addr>"
sd_app="10.137.0.50"
sd_mon="10.137.0.51"

iptables -I FORWARD 2 -s "$sd_dev" -d "$sd_app" -j ACCEPT
iptables -I FORWARD 2 -s "$sd_dev" -d "$sd_mon" -j ACCEPT
iptables -I FORWARD 2 -s "$sd_app" -d "$sd_mon" -j ACCEPT
iptables -I FORWARD 2 -s "$sd_mon" -d "$sd_app" -j ACCEPT
```

Run those commands on `sys-firewall` with

```bash
sudo sh /rw/config/qubes-firewall-user-script
```

Now from `sd-dev`, you should be able to do

```bash
ssh sdadmin@10.137.0.50
```

and log in with the password `securedrop`.

### 74.6.3 SSH using keys

**Tip**: You likely already have an SSH keypair configured for access to GitHub. If not, create one with `ssh-keygen -b 4096 -t rsa`. The configuration logic will use the key at `~/.ssh/id_rsa` to connect to the VMs.

Later we’ll be using Ansible to provision the application VMs, so we should make sure we can SSH between those machines without needing to type a password. On `sd-dev`:

```bash
ssh-copy-id sdadmin@10.137.0.50
ssh-copy-id sdadmin@10.137.0.51
```

Confirm that you’re able to ssh as user `sdadmin` from `sd-dev` to `sd-mon-base` and `sd-app-base` without being prompted for a password.

### 74.7 SecureDrop Installation

We’re going to configure `sd-dev` to build the SecureDrop `.deb` files, then we’re going to build them, and provision `sd-app` and `sd-mon`. Follow the instructions in the developer documentation to set up the development environment.

Once finished, build the Debian packages for installation on the staging VMs.

```bash
make build-debs
```

The `.deb` files will be available in `build/`.

### 74.8 Managing Qubes RPC for Admin API capability

We’re going to be running Qubes management commands on `sd-dev`, which requires some additional software. Install it with
sudo apt install qubes-core-admin-client

You’ll need to grant the sd-dev VM the ability to create other VMs, by editing the Qubes RPC policies in dom0. Here is an example of a permissive policy, sufficient to grant sd-dev management capabilities over VMs it creates:

```
/etc/qubes-rpc/policy/include/admin-local-rwx:
    sd-dev $tag:created-by-sd-dev allow,target=$adminvm

/etc/qubes-rpc/policy/include/admin-global-rwx:
    sd-dev $adminvm allow,target=$adminvm
    sd-dev $tag:created-by-sd-dev allow,target=$adminvm
```

**Tip:** See the Qubes documentation for details on leveraging the Admin API.

### 74.9 Creating staging instance

After creating the StandaloneVMs as described above:

- sd-dev
- sd-staging-base
- sd-app-base
- sd-mon-base

And after building the SecureDrop .debs, we can finally provision the staging environment. In from the root of the SecureDrop project in sd-dev, run:

```
make staging
```

One limitation of Qubes is that the reboot handlers which run during provisioning can shut down the VMs, but not start them again. When you see the message **RUNNING HANDLER [common : Wait for server to come back.]** you must start the VMs again manually from dom0 with the command `qvm-start sd-app && qvm-start sd-mon`.

The `make staging` command invokes the `qubes-staging` Molecule scenario. You can also run constituent Molecule actions directly, rather than using the Makefile target:

```
molecule create -s qubes-staging
molecule converge -s qubes-staging
molecule test -s qubes-staging
```

**Note:** If the Molecule converge scenario fails with an error like "stderr": "app: Failed to clone appmenus, qvm-appmenus missing\" you may be running into a bug in Qubes that prevents non-dom0 VMs from cloning new VMs. A workaround is described here.

That’s it. You should now have a running, configured SecureDrop staging instance running on your Qubes machine. For day-to-day operation, you should run sd-dev in order to make code changes, and use the Molecule commands above to provision staging VMs on-demand. To remove the staging instance, use the Molecule command:

```
molecule destroy -s qubes-staging
```
Targeting Xenial support

The SecureDrop project has always recommended Ubuntu Trusty for the server operating system. In April 2019, the Long-Term Support (LTS) status for Trusty will end. We plan to migrate to Ubuntu Xenial 16.04, which will be supported until April 2021.

In order to evaluate how to support Xenial as we prepare for the transition, we’ve created a developer environment suitable for provisioning VMs based on Xenial.

### 75.1 Running the Xenial dev env

If you’re using the libvirt Vagrant provider, you’ll need a libvirt-format Xenial base image. To set one up, run the following commands:

```bash
vagrant box add --provider virtualbox bento/ubuntu-16.04
vagrant mutate bento/ubuntu-16.04 mutate libvirt
```

Due to packaging logic changes, you’ll need to build the Debian packages with overrides enabled for Xenial support. Then run the Xenial scenario.

```bash
make build-debs-xenial
make staging-xenial
```

The VMs will now be available. Depending your choice of VM provider, you can log into the machines with the following commands:

### 75.1.1 libvirt:

```bash
molecule login -s libvirt-staging-xenial -h app-staging
```
75.1.2 virtualbox:

```
molecule login -s virtualbox-staging-xenial -h app-staging
```

To run the testinfra tests against the Xenial environment, you can use the commands:

75.1.3 libvirt:

```
molecule verify -s libvirt-staging-xenial
```

75.1.4 virtualbox:

```
molecule verify -s virtualbox-staging-xenial
```

If you encounter errors, re-running the `make staging-xenial` target may help. Naturally, we want the process to be error-free reliably.

75.2 Further reading

More detailed research notes on evaluating Xenial support can be found in the following GitHub issues:

- #3207 - [xenial] Perform timeboxed install attempt of SecureDrop against Ubuntu 16.04
- #3491 - [xenial] Perform timeboxed upgrade attempt of SecureDrop from Ubuntu 14.04 to 16.04
- #3204 - Migrate SecureDrop servers to Ubuntu 16.04 (Xenial)
This document outlines the threat model for SecureDrop 0.3 and is inspired by a document Adam Langley wrote for Pond. The threat model is defined in terms of what each possible adversary can achieve. This document is always a work in progress. If you have questions or comments, please open an issue on GitHub or send an email to securedrop@freedom.press.

76.1 Actors

The SecureDrop ecosystem comprises a host of actors, organized by the following high-level categories: Users, Adversaries, and Systems.

76.1.1 Users

The following table of the users who interact with the SecureDrop web application. Note that the airgapped SVS with the GPG Submission Key is required to decrypt submissions or messages.

<table>
<thead>
<tr>
<th>User Type</th>
<th>Trust Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>• Submit a document or message</td>
</tr>
<tr>
<td>Recurring source</td>
<td>• Submit another document or message</td>
</tr>
<tr>
<td></td>
<td>• Read replies</td>
</tr>
<tr>
<td>Journalist</td>
<td>• Download all gpg-encrypted documents from all sources</td>
</tr>
<tr>
<td></td>
<td>• Download all gpg-encrypted messages from all sources</td>
</tr>
<tr>
<td></td>
<td>• Reply to all sources</td>
</tr>
<tr>
<td>Admin</td>
<td>• Download all gpg-encrypted documents from all sources</td>
</tr>
<tr>
<td></td>
<td>• Download all gpg-encrypted messages from all sources</td>
</tr>
<tr>
<td></td>
<td>• Reply to all sources</td>
</tr>
<tr>
<td></td>
<td>• Change the SecureDrop instance logo</td>
</tr>
<tr>
<td></td>
<td>• SSH and root privileges on app and mon servers</td>
</tr>
</tbody>
</table>
76.1.2 Adversaries

We consider the following classes of attackers for the design and assessment of SecureDrop:

<table>
<thead>
<tr>
<th>Adversary</th>
<th>Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nation State / Law Enforcement / Global Adversary</td>
<td>• Large scale, full-packet network capture</td>
</tr>
<tr>
<td></td>
<td>• Active network attacks</td>
</tr>
<tr>
<td></td>
<td>• Advanced attacks on infrastructure</td>
</tr>
<tr>
<td></td>
<td>• Hardware and software implants for persistence</td>
</tr>
<tr>
<td></td>
<td>• Cryptanalysis</td>
</tr>
<tr>
<td></td>
<td>• Exploitation of unknown vulnerabilities</td>
</tr>
<tr>
<td>Large Corporation</td>
<td>• Limited network capture</td>
</tr>
<tr>
<td></td>
<td>• Some targeted attacks on infrastructure</td>
</tr>
<tr>
<td></td>
<td>• Use of known vulnerabilities</td>
</tr>
<tr>
<td></td>
<td>• Mostly limited to software-based attacks</td>
</tr>
<tr>
<td>Internet Service Provider</td>
<td>• Full network capture</td>
</tr>
<tr>
<td></td>
<td>• Mostly limited to network-based attacks</td>
</tr>
<tr>
<td>User Error</td>
<td>• Source, Journalist, Administrator or Developer error</td>
</tr>
<tr>
<td>Dedicated Individual</td>
<td>• Use of known vulnerabilities</td>
</tr>
<tr>
<td></td>
<td>• Mostly limited to software-based attacks</td>
</tr>
</tbody>
</table>

76.1.3 Systems

For more information about the various systems involved in a SecureDrop deployment, please visit the hardware section.

<table>
<thead>
<tr>
<th>System</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware Firewall</td>
<td>• Dedicated Hardware Firewall</td>
</tr>
<tr>
<td></td>
<td>• pfSense-based</td>
</tr>
<tr>
<td></td>
<td>• 3 Interfaces: <code>app</code>, <code>mon</code> and <code>admin</code></td>
</tr>
<tr>
<td>Application Server</td>
<td>• SecureDrop Source Interface</td>
</tr>
<tr>
<td></td>
<td>• SecureDrop Journalist Interface</td>
</tr>
<tr>
<td></td>
<td>• SSH Server</td>
</tr>
<tr>
<td></td>
<td>• Ossec Client</td>
</tr>
<tr>
<td>Monitor Server</td>
<td>• Ossec Server</td>
</tr>
<tr>
<td></td>
<td>• SSH Server</td>
</tr>
<tr>
<td>Journalist/Admin Workstation</td>
<td>• Internet-connected laptop</td>
</tr>
<tr>
<td></td>
<td>• Tails USB with persistence volume</td>
</tr>
<tr>
<td>Secure Viewing Station (SVS)</td>
<td>• Airgapped and stripped-down laptop</td>
</tr>
<tr>
<td></td>
<td>• Tails USB with persistence volume</td>
</tr>
</tbody>
</table>

76.2 Assumptions

The following assumptions are accepted in the threat model of every SecureDrop project:

76.2.1 Assumptions About the Source

- The source acts reasonably and in good faith, e.g. if the source were to give their credentials or private key material to the attacker that would be unreasonable.
- The source would like to remain anonymous, even against a forensic attacker.
- The source obtains an authentic copy of Tails and the Tor Browser.
- The source follows our guidelines for using SecureDrop.
- The source is accessing an authentic SecureDrop site.

76.2.2 Assumptions About the Admin and the Journalist

- The admin and the journalist act reasonably and in good faith, e.g. if either of them were to give their credentials or private key material to the attacker that would be unreasonable.
- The admin and the journalist obtain authentic copies of Tails.
- The journalist follows our guidelines for using SecureDrop and working with submitted documents.

76.2.3 Assumptions About the Person Installing SecureDrop

- This person (usually the admin) acts reasonably and in good faith, e.g. if they were to give the attacker system-level access that would be unreasonable.
- The person obtains an authentic copy of SecureDrop and its dependencies.
- The person follows our guidelines for deploying the system, setting up the landing page for the organization, and for installing SecureDrop.

76.2.4 Assumptions About the Source’s Computer

- The computer correctly executes Tails or the Tor Browser.
- The computer is not compromised by malware.

76.2.5 Assumptions About the Admin Workstation and the Journalist Workstation

- The computer correctly executes Tails.
- The computer and the Tails device are not compromised by malware.
- The two-factor authentication device used with the workstation are not compromised by malware.

76.2.6 Assumptions About the Secure Viewing Station

- The computer is airgapped.
- The computer correctly executes Tails.
- The computer and the Tails device are not compromised by malware.

76.2.7 Assumptions About the SecureDrop Hardware

- The servers correctly execute Ubuntu, SecureDrop and its dependencies.
- The servers, network firewall, and physical media are not compromised by malware.
76.2.8 Assumptions About the Organization Hosting SecureDrop

• The organization wants to preserve the anonymity of its sources.
• The organization acts in the interest of allowing sources to submit documents, regardless of the contents of these documents.
• The users of the system, and those with physical access to the servers, can be trusted to uphold the previous assumptions unless the entire organization has been compromised.
• The organization is prepared to push back on any and all requests to compromise the integrity of the system and its users, including requests to deanonymize sources, block document submissions, or hand over encrypted or decrypted submissions.

76.2.9 Assumptions About the World

• The security assumptions of RSA (4096-bit GPG and SSH keys) are valid.
• The security assumptions of scrypt with randomly-generated salts are valid.
• The security/anonymity assumptions of Tor and the Hidden Service protocol are valid.
• The security assumptions of the Tails operating system are valid.
• The security assumptions of SecureDrop dependencies, specifically Ubuntu, the Linux kernel, application packages, application dependencies are valid.

76.2.10 Other Assumptions or Factors

• The level of press freedom may vary in both geography and time.
• The number of daily Tor users in a country can greatly vary.

76.3 Assets

<table>
<thead>
<tr>
<th>Asset Type</th>
<th>Asset</th>
</tr>
</thead>
</table>
| Assets relating to SecureDrop users | • Login details  
| | • Encryption key(s)  
| | • SSH details |
| Assets relating to the publicly accessed system | • Access to documents via server  
| | • Access to documents via Journalist Interface  
| | • Access to admin privileges via Journalist Interface  
| | • Access to user alerts, support tickets |
| Assets relating to the underlying system | • SecureDrop code manipulation  
| | • Dependency code manipulation |

76.4 Implications of SecureDrop Area Compromise

76.4.1 What a Compromise of the Application Server Can Surrender

• The server sees the plaintext codename, used as the login identifier, of every source.
SecureDrop Documentation, Release 1.0.0

• The server sees all HTTP requests made by the source, the admin, and the journalist.
• The server sees the plaintext submissions of every source.
• The server sees the plaintext communication between journalists and their sources.
• The server stores the Tor Hidden Service private key for the source interface.
• The server stores the Tor Hidden Service private key and ATHS token for the Journalist interface.
• The server stores and (optional) TLS private key and certificate (if HTTPS is enabled on the source interface)
• The server stores hashes of codenames, created with scrypt and randomly-generated salts.
• The server stores journalist password hashes, created with scrupt and randomly-generated salts, as well as TOTP seeds.
• The server stores only encrypted submissions and communication on disk.
• The server stores a GPG key for each source, with the source’s codename as the passphrase.
• The server may store plaintext submissions in memory for at most 24 hours.
• The server stores sanitized Tor logs, created using the SafeLogging option, for the Source Interface, the Journalist Interface, and SSH.
• The server stores both access and error logs for the Journalist Interface.
• The server stores connection history and audit logs for the admin.
• The server can connect to the Monitor Server using an SSH key and a passphrase.

76.4.2 What a Compromise of the Monitor Server Can Surrender

• The server stores the plaintext alerts on disk, data may also reside in RAM.
• The server stores the GPG public key the OSSEC alerts are encrypted to.
• The server stores plaintext credentials for the SMTP relay used to send OSSEC alerts.
• The server stores the email address the encrypted OSSEC alerts are sent to.
• The server stores sanitized Tor logs, created using the SafeLogging option, for SSH.
• The server stores connection history and audit logs for the admin.
• The server stores OSSEC and Procmail logs on disk.
• The server can connect to the Application Server using an SSH key and a passphrase.

76.4.3 What a Compromise of the Workstations Can Surrender

• The Admin Workstation requires Tails with a persistent volume, which stores information such as GPG and SSH keys, as well as a database with passphrases for the Application Server, the Monitor Server, and the GPG key the Monitor Server will encrypt OSSEC alerts to.
• The Journalist Workstation requires Tails with a persistent volume, which stores information such as the Hidden Service value required to connect to the Journalist Interface, as well as a database with passphrases for the Journalist Interface.
• The Secure Viewing Station requires Tails with a persistent volume, which stores information such as the SecureDrop application’s GPG key, as well as a database with the passphrase for that key.

76.4. Implications of SecureDrop Area Compromise
76.4.4 What a Compromise of the Source’s Property Can Surrender

- Use of the Tor Browser will leave traces that can be discovered through a forensic analysis of the source’s property following either a compromise or physical seizure. Unless the compromise or seizure happens while the source is submitting documents to SecureDrop, the traces will not include information about sites visited or actions performed in the browser.

- Use of Tails with a persistent volume will leave traces on the device the operating system was installed on. Unless the compromise or seizure happens while the source is submitting documents to SecureDrop, or using the persistent volume, the traces will not include information about sites visited or actions performed in the browser or on the system.

- SecureDrop 0.3 encourages sources to protect their codenames by memorizing them. If a source cannot memorize the codename right away, we recommend writing it down and keeping it in a safe place at first, and gradually working to memorize it over time. Once the source has memorized it, they should destroy the written copy. If the source does write down the codename, a compromise or physical seizure of the source’s property may result in the attacker obtaining the source’s codename.

- An attacker with access to the source’s codename can:
  - Show that the source has visited the SecureDrop site, but not necessarily submitted anything.
  - Upload new documents or submit messages.
  - Communicate with the journalist as that source.
  - See any replies from journalists that the source has not yet deleted.

76.4.5 What a Physical Seizure of the Source’s Property Can Surrender

- Document use of Tor or Tails, but not necessarily research into SecureDrop

- Prevent the source from submitting documents by taking the device the documents are stored on.

- If the property is seized while powered on, the attacker can also analyze any plaintext information that resides in RAM.

- Tamper with the hardware.

- A physical seizure of, and access to, the source’s codename will allow the attacker to access the Source Interface as that source.

- A physical seizure of the admin’s property will allow the attacker to:
  - Prevent the admin from working on SecureDrop for some period of time.
  - Access any stored, decrypted documents taken off the Secure Viewing Station.
  - If the property is seized while powered on, the attacker can also analyze any plaintext information that resides in RAM.

- A physical seizure of, and access to, the admin’s Tails persistent volume, password database, and two-factor authentication device will allow the attacker to access both servers and the Journalist Interface.

76.4.6 What Compromise of the Admin’s Property Can Surrender

- To access the Journalist Interface, the Application Server, or the Monitor Server, the attacker needs to obtain the admin’s login credentials and the admin’s two-factor authentication device. Unless the attacker has physical access to the servers, the attacker will also need to obtain the Hidden Service values for the Interface and the servers. This information is stored in a password-protected database in a persistent volume on the admin’s Tails
device. The volume is protected by a passphrase. If the admin’s two-factor authentication device is a mobile phone, this will also be protected by a passphrase.

- An attacker with access to the admin’s computer can:
  - Access any stored, decrypted documents taken off the Secure Viewing Station.

- An attacker with access to the persistent volume on the admin’s Tails device can:
  - Add, modify, and delete files on the volume.
  - Access the Hidden Service values used by the Interfaces and the servers.
  - Access SSH keys and passphrases for the Application Server and the Monitor Server.
  - Access the GPG key and passphrase for the encrypted OSSEC email alerts.
  - Access the credentials for the account the encrypt alerts are sent to.
  - Access the admin’s personal GPG public key, if stored there.

- An attacker with admin access to the Journalist Interface can:
  - Add, modify, and delete journalist users.
  - Change the codenames associated with sources within the Interface.
  - Download, but not decrypt, submissions.
  - Communicate with sources.
  - Delete one or more submissions.
  - Delete one or more sources, which destroys all communication with that source and prevents the source from ever logging back in with that codename.

- An attacker with admin access to the Application Server can:
  - Add, modify, and delete software, configurations, and other files.
  - See all HTTP requests made by the source, the admin, and the journalist.
  - See the plaintext codename of a source as they are logging in.
  - See the plaintext communication between a source and a journalist as it happens.
  - See the stored list of hashed codenames.
  - Access the GPG public key used to encrypt communications between a journalist and a source.
  - Download stored, encrypted submissions and replies from the journalists.
  - Decrypt replies from the journalists if the source’s codename, and thus the passphrase, is known.
  - Analyze any plaintext information that resides in RAM, which may include plaintext of submissions made within the past 24 hours.
  - Review logs stored on the system.
  - Access the Monitor Server.

- An attacker with admin access to the Monitor Server can:
  - Add, modify, and delete software, configurations, and other files.
  - Change the SMTP relay, email address, and GPG key used for OSSEC alerts.
  - Analyze any plaintext information that resides in RAM.
  - Review logs stored on the system.
– Trigger arbitrary commands to be executed by the OSSEC agent user, which, assuming the attacker is able to escalate privileges, may affect the Application Server.

76.4.7 What a Physical Seizure of the Admin’s Property Can Achieve

• Tamper with the hardware.
• Prevent the admin from working on SecureDrop for some period of time.
• Access any stored, decrypted documents taken off the Secure Viewing Station.
• If the property is seized while powered on, the attacker can also analyze any plaintext information that resides in RAM.
• A physical seizure of, and access to, the admin’s Tails persistent volume, password database, and two-factor authentication device will allow the attacker to access both servers and the Journalist Interface.

76.4.8 What a Compromise of the Journalist’s Property Can Achieve

• To access the Journalist Interface, the attacker needs to obtain the journalist’s login credentials and the journalist’s two-factor authentication device or seed. Unless the attacker has physical access to the server, the attacker will also need to obtain the Hidden Service value for the Interface. This information is stored in a password-protected database in a persistent volume on the journalist’s Tails device. The volume is protected by a passphrase. If the journalist’s two-factor authentication device is a mobile phone, this will also be protected by a passphrase.
• An attacker with access to the journalist’s computer can:
  – Access any stored, decrypted documents taken off the Secure Viewing Station.
• An attacker with access to the persistent volume on the journalist’s Tails device can:
  – Add, modify, and delete files on the volume.
  – Access the Hidden Service values used by the Journalist Interface.
  – Access SSH keys and passphrases for the Application Server and the Monitor Server.
• An attacker with journalist access to the Journalist Interface can:
  – Change the codenames associated with sources within the interface.
  – Download, but not decrypt, submissions.
  – Delete one or more submissions.
  – Communicate with sources.
  – If the journalist has admin privileges on SecureDrop, they can create new journalist accounts.

76.4.9 What a Physical Seizure of the Journalist’s Property Can Achieve

• Tamper with the hardware.
• Prevent the journalist from working on SecureDrop for some period of time.
• Access any stored, decrypted documents taken off the Secure Viewing Station.
• If the property is seized while powered on, the attacker can also analyze any plaintext information that resides in RAM.
• A physical seizure of, and access to, the journalist’s Tails persistent volume, password database, and two-factor authentication device will allow the attacker to access the Journalist Interface.

76.4.10 What a Compromise of the Application Server Can Achieve

• If the Application Server is compromised, the system user the attacker has control over defines what kind of information the attacker will be able to view and what kind of actions the attacker can perform.

• An attacker with access to the debian-tor user can:
  – View, modify, and delete all files owned by this user. This includes sanitized Tor logs, created using the SafeLogging option, for SSH, the Source Interface and the Journalist Interface.
  – View, modify, and delete the Tor configuration file, root is required to reload the config.

• An attacker with access to the ossec user can:
  – Add, view, modify, and delete the log files, and in doing so send inaccurate information to the Monitor Server and the admin.

• An attacker with access to the www-data user can:
  – View, modify, and delete all files owned by this user. This includes all files in use by the SecureDrop application, such as text, code, the database containing encrypted submissions and communications. The attacker needs root access to reload configuration files.
  – View, modify, and delete both access and error logs for the Journalist Interface.
  – View any HTTP requests made by the source, the admin, and the journalist in that moment. This includes seeing plaintext codenames, submissions, and communications.
  – Add and delete communications between a journalist and a source by writing to the database.

• An attacker with access to the root user can:
  – Do anything the www-data user can do in terms of the SecureDrop application, this user is in full control of the server and can view, modify, and delete anything at will. This user is not able to decrypt submissions or communications, unless the attacker has access to the encryption key required to do so.

76.4.11 What a Physical Seizure of the Application Server Can Achieve

• If the Application Server is seized, the attacker will be able to view any and all unencrypted files on the server. An attacker will be able to modify any and all files on the server. This includes all files in use by the SecureDrop Application. If the server is seized while it is powered on, the attacker can also analyze any plaintext information that resides in RAM. The attacker can also tamper with the hardware.

76.4.12 What a Compromise of the Monitor Server Can Achieve

• If the Monitor Server is compromised, the system user the attacker has control over defines what kind of information the attacker will be able to view and what kind of actions the attacker can perform.

• An attacker with access to the debian-tor user can:
  – View, modify, and delete all files owned by this user. This includes sanitized Tor logs, created using the SafeLogging option, for SSH.
  – View, modify, and delete the Tor configuration file, root is required to reload the config.

• An attacker with access to the ossec user can:
– View all ossec logs and alerts on disk.
– Modify the ossec configuration.
– Send (or suppress) emails to administrators and journalists.

• An attacker with access to the root user can:
  – Do anything the ossec user can do in terms of the SecureDrop application, this user is in full control of the server and can view, modify, and delete anything at will. This user is not able to decrypt encrypted email alerts, unless the attacker has access to the encryption key required to do so.

76.4.13 What a Physical Seizure of the Monitor Server Can Achieve

• If the Monitor Server is seized, the attacker will be able to view any and all unencrypted files on the server. This includes all files in use by OSSEC. If the server is seized while it is powered on, the attacker can also analyze any plaintext information that resides in RAM. The attacker can also tamper with the hardware.
• If the Monitor Server is no longer online or tampered with, this will have an effect on the quantity and accuracy of notifications sent to admins or journalists.

76.4.14 What a Compromise of the Secure Viewing Station Can Achieve

• The Secure Viewing Station is only useful to an attacker while powered on and with the Tails persistent volume mounted. The attacker may learn more if the Transfer Device or the Export Device are in use at the time of compromise or seizure. A physical seizure of this machine, its Tails device, the Transfer Device or the Export Device will also achieve nothing, assuming that the Tails and VeraCrypt implementations of full-disk encryption work as expected.

• A compromise of the Secure Viewing Station allows the attacker to:
  – Run commands as the amnesia user.
  – View, modify, and delete files owned by the amnesia user. This includes the Submission Private Key used to encrypt and decrypt submitted documents.
  – View, modify, and delete submissions in encrypted form
  – View, modify, and delete decrypted submissions, if they are stored in decrypted form on the Secure Viewing Station, or if the Export Device is in use.
  – Export the Submission Private Key key (unless there is a passphrase set).

76.4.15 What a Physical Seizure of the Secure Viewing Station Can Achieve

• The Secure Viewing Station is only useful to an attacker while powered on and with the Tails persistent volume mounted. The attacker may learn more if the Transfer Device or the Export Device are in use at the time of compromise or seizure. A physical seizure of this machine, its Tails device, the Transfer Device or the Export Device will also achieve nothing, assuming that the Tails and VeraCrypt implementations of full-disk encryption work as expected.

• A physical seizure of the Secure Viewing Station, while on and with the persistent volume decrypted and mounted, allows the attacker to:
  – Tamper with the hardware.
  – Run commands as the amnesia user.
  – View, modify, and delete the Submission Private Key used to encrypt and decrypt submitted documents.
– View, modify, and delete decrypted submissions, if they are stored in decrypted form on the Secure Viewing Station, or if the Export Device is in use.

76.4.16 What a Local Network Attacker Can Achieve Against the Source, Admin, or Journalist:

- A local network can observe when they are using Tor.
- A local network can block Tor and prevent them from accessing SecureDrop.
- A local network may be able to deduce use of SecureDrop by looking at request sizes, plaintext uploads and encrypted downloads, although research suggests this is very difficult.

76.4.17 What a Global Adversary Can Achieve Against the Source, Admin, or Journalist:

- A global adversary capable of observing all Internet traffic may have more luck than the local network attacker in deducing use of SecureDrop by looking at request sizes, plaintext uploads and encrypted downloads.
- A global adversary may be able to link a source to a specific SecureDrop server.
- A global adversary may be able to link a source to a specific journalist.
- A global adversary may be able to correlate data points during a leak investigation, including looking at who has read up on SecureDrop and who has used Tor.
- A global adversary may be able to forge an SSL certificate and use it to spoof an organization’s HTTPS Landing Page, thereby tricking the source into visiting a fake SecureDrop site.

76.4.18 What a Random Person on the Internet Can Achieve

- A random person can attempt to DoS the SecureDrop server and overwhelm the journalists by generating a high number of codenames and uploading many large documents.
- A random person can submit empty, forged, or inaccurate documents.
- A random person can submit malicious documents, e.g. malware that will attempt to compromise the Secure Viewing Station.
- A random person can attempt to get sensitive information from a SecureDrop user’s browser session, such as the source’s codename.
- A random person can attempt to compromise the SecureDrop server by attacking the exposed attack surface, including the kernel network stack, Tor, Apache, the SecureDrop web interfaces, Python, OpenSSH, and the TLS implementation.
The following diagram captures all data flows to and from a SecureDrop deployment.
SecureDrop is a complex ecosystem comprised of various pieces of hardware, a diverse codebase, multiple user roles, and varied software dependencies. As such, an adversary can compromise any one of these components through a variety of attacks, as detailed below. We’ve categorized attacks and countermeasures by SecureDrop architecture area for clarity.

There are certain attacks that cannot be mitigated by any of the technical or operational countermeasures built into SecureDrop. Attacks of a political nature — for example, if a source, journalist, or organization is threatened with legal action — are context-dependent, and determined by an ever-shifting climate around press freedoms. While these attack vectors are out of the scope of this document, they should be factored in to any organization’s threat model with regional and political specificity.

### 78.1 Application Code — SecureDrop Repository/Release

#### 78.1.1 Attacks to the Application Code — SecureDrop Repository/Release

- Malicious code introduced in SecureDrop repository
- Malicious code introduced in SecureDrop release
- Failure to encrypt submissions as they are written to disk

#### 78.1.2 Countermeasures on the Application Code — SecureDrop Repository/Release

- Code (git tags) and releases (packages uploaded to apt) are signed with the airgapped signing key
- Protection is placed on master and develop branch on GitHub
- For SecureDrop Developers, 2-factor authentication is mandated on GitHub
- Community trust is built through 3 trusted code owners and code reviews
78.2 Application Code — Source Interface and Journalist Interface

78.2.1 Attacks to the Application Code — Source Interface and Journalist Interface

- Configuration vulnerability in Source or Journalist Interface
- Lack of segmentation between Source and Journalist Interface
- Session management vulnerability in Source or Journalist Interface
- Malicious input vulnerability in Source or Journalist Interface
- Configuration vulnerability in Source or Journalist Interface
- Authentication vulnerability in Source or Journalist Interface
- Access control vulnerability in Source or Journalist Interface
- Data protection vulnerability in Source or Journalist Interface
- Communications vulnerability in Source or Journalist Interface
- Error handling and logging vulnerability in Source or Journalist Interface
- HTTP security configuration vulnerability in Source or Journalist Interface
- File and resource vulnerability in Source or Journalist Interface
- Business logic vulnerability in Source or Journalist Interface
- Web services vulnerability in Source or Journalist Interface

78.2.2 Countermeasures on both Source and Journalist Interfaces

- Interfaces run on an end-to-end encrypted Tor Onion Service
- Sensitive source and submission data is sent through HTTP POST
- All source submissions are encrypted with GPG at rest using the airgapped Submission Key
- Interface sessions are invalidated after a user logs out or inactivity over 120 minutes
- Session control on Interface includes CSRF token in Flask Framework
- All Interface session data (except language and locale selection) is discarded at logout, and fully deleted upon exiting the Tor Browser
- A number of mitigations are in place as protection against malicious input vulnerabilities on the Source and Journalist Interfaces:
  - X-XSS-PROTECTION is enabled
  - Content-Security-Policy is set to self
  - SQLAlchemy is used as ORM for all database queries
  - Application does not execute uploaded submission data
- A number of mitigations are in place as protection against the risk of an HTTP misconfiguration on the Source and Journalist Interfaces:
  - Cache control header is set to “no store;”
  - HTTP headers do not expose version information of system components
  - X-Content-Type is set to “nosniff;”
SecureDrop Documentation, Release 1.0.0

– Content-Security-Policy is set to “self;”
– X-XSS-Protection is set to “1”

78.2.3 Countermeasures unique to Source Interface

• TLS on Source Interface is opt-in with an EV cert
• Only HTTP GET, POST and HEAD methods are allowed
• A number of mitigations are in place as protection against access control vulnerabilities on the Source Interface:
  – Source codenames are long and automatically generated
  – Hashed codenames are stored in a database hashed with a unique salt
  – Source codename reset functionality is not available
  – Source login does not display information about prior submissions
  – Source login requires 7-word codename to check Source Interface for replies

78.2.4 Countermeasures unique to Journalist Interface

• Journalist Interface is located behind an authenticated Onion Service and only privileged users have required authorization token
• Only HTTP GET, POST, HEAD and DELETE methods are allowed
• A number of mitigations are in place as protection against access control vulnerabilities on the Journalist Interface:
  – Apache autoindex module is disabled
  – Journalist/Admin passphrases are long and automatically generated
  – Passphrases are stored in a database hashed with a unique salt
  – Account generation/revocation/reset is restricted to Admin role
  – Two-factor authentication is required through a TOTP token or a Yubikey

78.3 Application Server and Monitor Server

78.3.1 Attacks on the Application Server and Monitor Server

• Application or Monitor Server configuration error
• Source or Journalist Interface is framed
• Application or Monitor Server is compromised
• Attacker exploits postfix
• Known vulnerabilities in the Linux kernel or packages used by app/mon servers
78.3.2 Countermeasures on Both Application and Monitor Servers

- Grsecurity/PaX linux patches prevent the exploitation of certain memory-corruption attacks
- AppArmor profiles further reduce process capabilities through Mandatory Access Control
- All SecureDrop infrastructure is provisioned via infrastructure-as-code (Ansible scripts)
- A cron job ensures that automatic nightly security updates are applied for OS packages
- Journalist Interface uses ATHS cookie
- Monitor Server should only expose SSH via Tor Onion Service. All other traffic should be blocked by firewall

78.3.3 Countermeasures Unique to Application Server

- SecureDrop Source and Journalist Interfaces uses X-Frame-Options: DENY header
- Browser Same Origin Policy should prevent the SecureDrop page from trivial modifications, but more complex attacks are mitigated via the X-Frame-Options: DENY HTTP header

78.3.4 Countermeasures Unique to Monitor Server

- OSSEC is used for intrusion detection/file integrity monitoring, and are sent to Admins via end-to-end encrypted email

78.4 SecureDrop Dependencies — Python, Tor, Linux Kernel, apt, Tails, Ubuntu, or Hardware Firewall Vulnerabilities

78.4.1 Attacks on SecureDrop Dependencies

- Known vulnerabilities in Python or libraries used by SecureDrop
- Known vulnerabilities in Tor (incl. Onion Service cryptography, authentication)
- Malicious apt package installed at install-time or during updates
- Known weakness in Onion Service cryptography
- Github is compromised
- Firewall is not up-to-date
- Tails ISO malicious
- Ubuntu ISO malicious
- Tor apt repo compromised
- Ubuntu apt repo compromised
- Tor Browser exploit
- Vulnerabilities/Compromise of Hardware Firewall
78.4.2 Countermeasures Against Vulnerabilities in Python or Libraries

- FPF performs vulnerability management for all Python packages used by SecureDrop
- CI will run safety check to ensure dependencies do not have a CVE associated with the version

78.4.3 Countermeasures Against Vulnerabilities in Tor

- A cron job ensures that automatic nightly security updates are applied for OS packages, including Tor
- Grsecurity/PaX linux patches prevent the exploitation of certain memory-corruption attacks
- AppArmor profiles further reduce process capabilities through Mandatory Access Control
- Hidden Service authentication is used as a complementary authentication and only used for defense-in-depth/attack surface reduction

78.4.4 Countermeasures Against Malicious apt Installs

- apt does GPG signature verification of all packages as long as it’s not explicitly disabled

78.4.5 Countermeasures Against Malicious Tails or Ubuntu ISOs

- SecureDrop Admin Guide instructs Users/Admins to validate checksum/signatures of downloaded images

78.4.6 Countermeasures Against Vulnerabilities in the Hardware Firewall

- SecureDrop Admin Guide informs administrators to update the hardware firewall and provides a very restrictive policy for accessing the administrative interface (blocked on app and mon ports of the firewall).
- Alert emails are sent out to admins when there are critical pfSense vulnerabilities.
- Application and Monitor Servers use IPTables as host-based firewall for defense-in-depth
- All application traffic is over Tor Hidden services (end-to-end encrypted) and all software packages are signed. Only DNS and NTP are transmitted over HTTP (unauthenticated and in cleartext)

78.5 Network Infrastructure — FPF Infrastructure or Organization Corporate Network

78.5.1 Attacks on Network Infrastructure

- Landing Page source control is compromised
- Landing Page host is compromised
- Landing Page is framed or unavailable
- Landing Page DNS leaks from SecureDrop/leaks-related subdomain
- Communications vulnerability in Source or Journalist Interface
- DNS requests to news organization’s subdomain for SecureDrop Landing Page, Freedom.press, torproject.org Tor activity, SD submissions may be correlated
• SecureDrop.org is compromised
• User web traffic to SecureDrop Landing Page uses CDN and may be logged
• Tor network exploit
• apt server man-in-the-middle used to serve old or malicious packages
• SecureDrop apt servers are compromised, or apt server man-in-the-middle attack injects malicious packages
• News Organization network is compromised
• OSSEC and/or Journalist alert SMTP account credentials compromised
• OSSEC and/or Journalist alert private key compromised
• SMTP relay compromised
• Admin’s network is monitored

78.5.2 Countermeasures in FPF Infrastructure

• Builds are independently validated by multiple developers
• Release files containing hashes (MD5, SHA1, SHA256, SHA512) of package file and package hashes are signed with an airgapped GPG key
• Developer key list is published and GPG-signed with the directory key
• SecureDrop updates are packaged in a .deb file and served through FPF’s apt repo
• Source code is validated/verified before packaging and signing the .deb

78.5.3 Countermeasures in News Organization Corporate Network

• SecureDrop environment should be strictly segregated from corporate environment
• Most SecureDrop application traffic goes over Tor and as such is encrypted end-to-end
• Alert emails to Journalists and Admins are GPG-encrypted (but not signed) to provide confidentiality
• OSSEC alerts are scrubbed for sensitive contents (application data, server IPs)
• Documented deployment best practices provide instructions to strengthen Landing Page security and privacy

78.6 User Behavior and Hardware — SecureDrop Hardware Tampering or Failure in Operational Security

78.6.1 Attacks on User Behavior or Hardware

• Journalist corporate workstation seized/tampered/compromised
• Transfer device seized/stolen/lost
• Admin workstation backup stick is compromised
• Admin two-factor authentication device is lost or compromised
• Admin SSH Key is compromised
• SecureDrop installer misconfigures server/firewall hardware
- Source uses tor2web or employer/corporate device
- Source shares that they are using SecureDrop/leaking documents
- Journalist/Admin gets phished from a submission or otherwise breaks the SVS airgap with malware

### 78.6.2 Countermeasures in User Behavior Recommendations

- **Source Guide** gives instructions on best practices for the entire submission workflow
- Source interface banner suggests that user disables JS (high security settings in Tor Browser)
- **Journalist Guide** informs users of malware risks, the importance of strict compartmentalization of SecureDrop-related activities
- **SecureDrop Deployment Guide** gives best practices for proper administration of the SecureDrop system, and its public-facing properties like the Landing Page
- **Admin Guide** gives instructions for long-term maintenance of the technical properties of the SecureDrop system, as well as operations to support Journalists
- All Admin tasks are completed over Tor/Tor authenticated Onion Services after installation
- Any Journalist/Admin password/2FA token resets can only be done by an Admin with password-protected SSH capability or authenticated Onion Service credentials.
- Persistent storage on the Admin Workstation is protected with LUKS/dm-crypt encryption