# 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 ....................................... 10

## 2 Journalist Guide
2.1 Updating Your Workstation .......................................... 15
2.2 Creating a GPG Key ................................................... 16
2.3 Connecting to the Journalist Interface ............................. 16
2.4 Daily Journalist Alerts About Submissions ....................... 17
2.5 Interacting With Sources ............................................ 18
2.6 Moving Documents to the Secure Viewing Station .............. 21
2.7 Decrypting on the Secure Viewing Station ....................... 26
2.8 Working with Documents ............................................ 28
2.9 Encrypting and Moving Documents to the Journalist Workstation ........................................ 29
2.10 Decrypting and Preparing to Publish ............................ 31

## 3 Admin Guide
3.1 Responsibilities ..................................................... 33
3.2 Common Tasks ....................................................... 35
3.3 Frequently Asked Questions ........................................ 45

## 4 Passphrase Best Practices
4.1 General Best Practices ................................................ 47
4.2 For Sources .......................................................... 48
4.3 For Journalists/Admins ............................................... 48

## 5 Overview
5.1 Technical Summary ................................................... 49
5.2 Infrastructure ........................................................ 50
5.3 Operation ............................................................. 51

## 6 Terminology
6.1 Source ............................................................... 53
6.2 Journalist ............................................................. 53
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.3</td>
<td>Set Up SSH Keys</td>
<td>140</td>
</tr>
<tr>
<td>17</td>
<td>Install SecureDrop</td>
<td>141</td>
</tr>
<tr>
<td>17.1</td>
<td>Install Prerequisites</td>
<td>141</td>
</tr>
<tr>
<td>17.2</td>
<td>Localization of the Source Interface and Journalist Interface</td>
<td>142</td>
</tr>
<tr>
<td>17.3</td>
<td>Configure the Installation</td>
<td>142</td>
</tr>
<tr>
<td>17.4</td>
<td>Install SecureDrop Servers</td>
<td>143</td>
</tr>
<tr>
<td>18</td>
<td>Configure the Admin Workstation Post-Install</td>
<td>145</td>
</tr>
<tr>
<td>18.1</td>
<td>Auto-connect to the Authenticated Tor Hidden Services</td>
<td>145</td>
</tr>
<tr>
<td>19</td>
<td>Create an Admin Account on the Journalist Interface</td>
<td>147</td>
</tr>
<tr>
<td>20</td>
<td>Test the Installation</td>
<td>149</td>
</tr>
<tr>
<td>20.1</td>
<td>Test Connectivity</td>
<td>149</td>
</tr>
<tr>
<td>20.2</td>
<td>Sanity-Check the Installation</td>
<td>149</td>
</tr>
<tr>
<td>20.3</td>
<td>Test the Web Interfaces</td>
<td>150</td>
</tr>
<tr>
<td>21</td>
<td>Onboard Journalists</td>
<td>151</td>
</tr>
<tr>
<td>21.1</td>
<td>Determine Access Protocol for the Secure Viewing Station</td>
<td>151</td>
</tr>
<tr>
<td>21.2</td>
<td>Create a Journalist Tails USB</td>
<td>152</td>
</tr>
<tr>
<td>21.3</td>
<td>Set Up Automatic Access to the Journalist Interface</td>
<td>152</td>
</tr>
<tr>
<td>21.4</td>
<td>Add an account on the Journalist Interface</td>
<td>153</td>
</tr>
<tr>
<td>21.5</td>
<td>Verify Journalist Setup</td>
<td>153</td>
</tr>
<tr>
<td>22</td>
<td>Overview</td>
<td>155</td>
</tr>
<tr>
<td>23</td>
<td>Landing Page</td>
<td>157</td>
</tr>
<tr>
<td>23.1</td>
<td>URL and Location</td>
<td>157</td>
</tr>
<tr>
<td>23.2</td>
<td>HTTPS Only (No Mixed Content)</td>
<td>157</td>
</tr>
<tr>
<td>23.3</td>
<td>Perfect Forward Secrecy</td>
<td>158</td>
</tr>
<tr>
<td>23.4</td>
<td>SSL Certificate Recommendations</td>
<td>158</td>
</tr>
<tr>
<td>23.5</td>
<td>Do Not Use Third-Party Analytics, Tracking, or Advertising</td>
<td>158</td>
</tr>
<tr>
<td>23.6</td>
<td>Do Not Hyperlink .onion Addresses</td>
<td>159</td>
</tr>
<tr>
<td>23.7</td>
<td>Avoid Direct Links to SecureDrop.org</td>
<td>159</td>
</tr>
<tr>
<td>23.8</td>
<td>Apply Security Headers</td>
<td>159</td>
</tr>
<tr>
<td>23.9</td>
<td>Additional Apache Configuration</td>
<td>160</td>
</tr>
<tr>
<td>23.10</td>
<td>Further Security Considerations</td>
<td>161</td>
</tr>
<tr>
<td>23.11</td>
<td>Landing Page Content Suggestions</td>
<td>161</td>
</tr>
<tr>
<td>24</td>
<td>Minimum requirements for the SecureDrop environment</td>
<td>163</td>
</tr>
<tr>
<td>25</td>
<td>Whole Site Changes</td>
<td>165</td>
</tr>
<tr>
<td>25.1</td>
<td>Suggested</td>
<td>165</td>
</tr>
<tr>
<td>26</td>
<td>Sample SecureDrop Privacy Policy</td>
<td>167</td>
</tr>
<tr>
<td>26.1</td>
<td>Collection of Information From Sources</td>
<td>167</td>
</tr>
<tr>
<td>26.2</td>
<td>Collection of Information About Journalists' Use of SecureDrop</td>
<td>168</td>
</tr>
<tr>
<td>26.3</td>
<td>Data Security</td>
<td>168</td>
</tr>
<tr>
<td>26.4</td>
<td>Children Under 13</td>
<td>168</td>
</tr>
<tr>
<td>26.5</td>
<td>Changes to This Policy</td>
<td>168</td>
</tr>
<tr>
<td>26.6</td>
<td>Contact</td>
<td>168</td>
</tr>
<tr>
<td>27</td>
<td>Pre-Install Hardware Checklist</td>
<td>169</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td><strong>28 Promoting Your SecureDrop Instance</strong></td>
<td>171</td>
<td></td>
</tr>
<tr>
<td>28.1 Make a High Profile Announcement</td>
<td>171</td>
<td></td>
</tr>
<tr>
<td>28.2 Provide a Clear Link on Your Homepage</td>
<td>171</td>
<td></td>
</tr>
<tr>
<td>28.3 Provide Links at the Bottom of Your Articles</td>
<td>173</td>
<td></td>
</tr>
<tr>
<td>28.4 Create an Instructional Video on How to Access and Use Your SecureDrop</td>
<td>173</td>
<td></td>
</tr>
<tr>
<td>28.5 Regularly Share Your SecureDrop Landing Page on Social Media</td>
<td>173</td>
<td></td>
</tr>
<tr>
<td>28.6 Target Potential Whistleblowers with Advertising</td>
<td>174</td>
<td></td>
</tr>
<tr>
<td>28.7 Put an Advertisement in Your Physical Paper</td>
<td>174</td>
<td></td>
</tr>
<tr>
<td><strong>29 What Makes SecureDrop Unique</strong></td>
<td>177</td>
<td></td>
</tr>
<tr>
<td>29.1 No Third Parties that Can Secretly be Subpoenaed</td>
<td>177</td>
<td></td>
</tr>
<tr>
<td>29.2 Limits the Metadata Trail as Much as Possible</td>
<td>177</td>
<td></td>
</tr>
<tr>
<td>29.3 Encrypted and Air-Gapped</td>
<td>178</td>
<td></td>
</tr>
<tr>
<td>29.4 Protects Against Hackers</td>
<td>178</td>
<td></td>
</tr>
<tr>
<td>29.5 Free and Open Source Software</td>
<td>178</td>
<td></td>
</tr>
<tr>
<td><strong>30 Useful Logs</strong></td>
<td>179</td>
<td></td>
</tr>
<tr>
<td>30.1 Both servers</td>
<td>179</td>
<td></td>
</tr>
<tr>
<td>30.2 Application Server</td>
<td>179</td>
<td></td>
</tr>
<tr>
<td>30.3 Monitor Server</td>
<td>179</td>
<td></td>
</tr>
<tr>
<td><strong>31 OSSEC Guide</strong></td>
<td>181</td>
<td></td>
</tr>
<tr>
<td>31.1 Setting Up OSSEC Alerts</td>
<td>181</td>
<td></td>
</tr>
<tr>
<td>31.2 Troubleshooting</td>
<td>184</td>
<td></td>
</tr>
<tr>
<td>31.3 Analyzing the Alerts</td>
<td>188</td>
<td></td>
</tr>
<tr>
<td><strong>32 Tails Guide</strong></td>
<td>189</td>
<td></td>
</tr>
<tr>
<td>32.1 Installing Tails on USB Sticks</td>
<td>189</td>
<td></td>
</tr>
<tr>
<td>32.2 Configure Tails for Use with SecureDrop</td>
<td>190</td>
<td></td>
</tr>
<tr>
<td><strong>33 Setting Up a Printer in Tails</strong></td>
<td>193</td>
<td></td>
</tr>
<tr>
<td>33.1 Installing and Printing via the Tails GUI</td>
<td>194</td>
<td></td>
</tr>
<tr>
<td>33.2 Troubleshooting</td>
<td>196</td>
<td></td>
</tr>
<tr>
<td>33.3 Printing from the Command Line</td>
<td>205</td>
<td></td>
</tr>
<tr>
<td><strong>34 HTTPS on the Source Interface</strong></td>
<td>207</td>
<td></td>
</tr>
<tr>
<td>34.1 Obtaining an HTTPS certificate for Onion URLs</td>
<td>207</td>
<td></td>
</tr>
<tr>
<td>34.2 Activating HTTPS in SecureDrop</td>
<td>209</td>
<td></td>
</tr>
<tr>
<td><strong>35 SSH Over Local Network</strong></td>
<td>211</td>
<td></td>
</tr>
<tr>
<td>35.1 Configuring SSH for Local Access</td>
<td>211</td>
<td></td>
</tr>
<tr>
<td><strong>36 SecureDrop On-Site Training Schedule</strong></td>
<td>213</td>
<td></td>
</tr>
<tr>
<td>36.1 Day 1: Preparation and Install</td>
<td>213</td>
<td></td>
</tr>
<tr>
<td>36.2 Day 2: Journalist and Admin Training</td>
<td>214</td>
<td></td>
</tr>
<tr>
<td><strong>37 Using a YubiKey with the Journalist Interface</strong></td>
<td>217</td>
<td></td>
</tr>
<tr>
<td>37.1 What is a YubiKey?</td>
<td>217</td>
<td></td>
</tr>
<tr>
<td>37.2 Download and Launch the YubiKey Personalization Tool</td>
<td>217</td>
<td></td>
</tr>
<tr>
<td>37.3 Setting Up Hardware-Based Codes</td>
<td>218</td>
<td></td>
</tr>
<tr>
<td>37.4 Adding Users</td>
<td>219</td>
<td></td>
</tr>
<tr>
<td>37.5 Using Your YubiKey</td>
<td>219</td>
<td></td>
</tr>
<tr>
<td><strong>38 Back Up, Restore, Migrate</strong></td>
<td>221</td>
<td></td>
</tr>
<tr>
<td>38.1 Minimizing Disk Use</td>
<td>221</td>
<td></td>
</tr>
<tr>
<td>38.2 Backing Up</td>
<td>222</td>
<td></td>
</tr>
</tbody>
</table>
47.2 Configuring SSH Over the Local Network ............................................. 275
47.3 Configuring Email Alerts About New Submissions ..... 275
47.4 Configuring Server Reboot Time ................................................... 275
47.5 Getting Support ................................................................. 275

48 Upgrade from 0.5.x to 0.6.x ......................................................... 277
48.1 Updating the Tails Workstations ................................................. 277
48.2 Troubleshooting Linux Kernel Issues ........................................... 277
48.3 Getting Support ................................................................. 282

49 Upgrade from 0.4.x to 0.5.x ........................................................ 283
49.1 Pull the Latest Release ......................................................... 283
49.2 Choose the List of Supported Languages ..................................... 284

50 Upgrade from 0.3.x to 0.4.x ......................................................... 285
50.1 Pull the Latest Release ......................................................... 285
50.2 Upgrade the Tails Persistence Configuration ................................ 286
50.3 Verify the Upgrades ............................................................. 286

51 Upgrade Tails from 1.x to 2.x ......................................................... 289
51.1 Upgrade Each Tails Device .................................................... 290
51.2 Finishing Up ...................................................................... 299
51.3 Troubleshooting ................................................................. 300

52 Upgrade Tails from 2.x to 3.x ......................................................... 303
52.1 Why You Should Upgrade ..................................................... 303
52.2 What You Need ................................................................. 303
52.3 1. Prepare the Master Tails USB ............................................. 304
52.4 2. Backup the Tails Drives .................................................... 304
52.5 3. Upgrade the Tails Drives .................................................... 315
52.6 4. Upgrade KeePassX Database ............................................. 320
52.7 5. Upgrade the 
Secure Viewing Station ......................................................... 321
52.8 6. Upgrade SecureDrop to 0.4.x ............................................. 321
52.9 If You Encounter Issues ....................................................... 321

53 Contributing to SecureDrop ............................................................. 323
53.1 Programmers .................................................................. 323
53.2 Writers .......................................................................... 324
53.3 DevOps ......................................................................... 325
53.4 Release Managers ............................................................. 325
53.5 Translators ......................................................................... 325
53.6 Designers ..................................................................... 325
53.7 Moderators and Support .................................................. 325

54 Setting Up the Development Environment ......................... 327
54.1 Prerequisites .................................................................. 327
54.2 Quick Start .................................................................... 327
54.3 Setting Up a Multi-Machine Environment ................................ 330

55 Making a PR to SecureDrop ......................................................... 333
55.1 Forking and Cloning the Project ............................................. 333
55.2 Make Your Changes and Push to the Fork ................................ 333
55.3 Making a Pull Request to Get Your Changes Merged in develop Branch ....... 334

56 Development of Securedrop-Admin in the Admin Directory .... 337
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 Slider that 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 Slider to Safest:

Click the Learn how to set it to Safest link in the warning banner and a message bubble will pop up explaining how to disable JavaScript and turn up the Slider:
It is recommended to move the Security Slider to Safest to protect your anonymity: Learn how to.

---

You appear to be using the Tor Browser. You can move the Security Slider to Safest in four steps.

1. Click the 🌐 Tor icon in the toolbar above
2. Click Security Settings...
3. Move the Slider to Safest, then click OK
4. Click here to refresh the page

---

Follow the instructions and the page should refresh automatically. Note that this will change the slider 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
Submit Documents 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 Practices](#).

Once you have generated a codename and put it somewhere safe, click **Continue**.
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.
Submit Files or Messages

If you are already familiar with GPG, you can optionally encrypt your files and messages with our public key before submission. Files are encrypted as they are received by SecureDrop. Learn more.

You can submit any kind of file, a message, or both.

Read Replies

There are no replies at this time.

Like all software, SecureDrop may contain security bugs. Use at your own risk.

Powered by SecureDrop 0.8.0-rc1.

Once you are finished submitting documents, be certain you have saved your secret codename and then click the Exit 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:

```
Confirm

Tor Browser will close all windows and tabs. All website sessions will be lost.

Restart Tor Browser now to reset your identity?

☐ Never ask me again
```

### 1.5 Continuing the Conversation

If you have already submitted a document and would like to check for responses, click the **Check for a Response** 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.
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.
Note: SecureDrop wants your feedback! Confused by something in our documentation? Let us know via an issue on GitHub or the community forum.

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, 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 add the Tails Administrator password, you will need to reboot to enable it.
2.2 Creating a GPG Key

Each journalist needs a personal GPG key for encrypting files. A GPG key has two parts: a public key and a private key. The private key, used for decryption, stays on the Journalist Workstation. The public key, used for encryption, is copied to the Secure Viewing Station.

If you do not yet have a GPG key, follow the instructions for your operating system to set one up:

- GNU/Linux
- Windows
- Mac OS

2.3 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. Your admin can help provide you with a Tails drive.

**Important:** See our guide on setting up *Tails for the Admin and Journalist Workstation* 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.

Each journalist has an authenticated Tor hidden service URL for logging in to the Journalist Interface. This must be done using the Tails operating system. Click the Journalist Interface icon on the desktop. This will open Tor Browser to a “.onion” page. Log in with your username, passphrase, and two-factor authentication token, as shown in the first screenshot below. (See *Using YubiKey with the Journalist Interface.*)

![Login to access the journalist interface]

 Powered by SecureDrop 0.8.0-rc1.
2.3.1 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.

2.4 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 her/his email and GPG public key to the admin. If a team of journalists 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.

2.5 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 button next to their current codename.

Tip: You can also Star interesting or promising sources to easily return to them later.

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 GPG public key. In order to read the messages or look at the documents you will need to transfer them to the Secure Viewing Station. 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.

2.5. Interacting With Sources
2.5.1 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 him or her. 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.

2.6 Moving Documents to the Secure Viewing Station

Documents sent by sources can only be viewed on the Secure Viewing Station. After clicking on an individual source, you will see the page below with any messages that source has sent you. 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 will appear asking if you want to Open or Save the file. Select Save File:
In order to protect you from malware, the browser in Tails will only allow you to download documents to a special persistent folder located at /home/amnesia/Tor Browser.
Tip: The special folder mentioned here is called **Tor Browser**, not “Persistent.” Attempting to download directly into the **Persistent** folder will only lead to frustration.

Once downloaded to this 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 will be known as your **Transfer Device**.
Eject the Transfer Device from the Journalist Workstation.

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 computer icon on your desktop, then on the Transfer Device. Drag and drop the file into your Persistent folder.

*Important:* Copy these documents to the Persistent folder before decrypting them. Otherwise you might accidentally decrypt the documents on the USB stick, and they could be recoverable in the future.
After successfully copying, erase the files from your *Transfer Device* by returning to the *Transfer Device* folder. Right-click on the files that need removal and click “Wipe” to securely delete the files from your device.

2.7 Decrypting on the *Secure Viewing Station*

To decrypt documents, return to your *Persistent* folder and double-click on the zipped file folder. After you extract the files, click on each file individually. A prompt will ask you for the application PGP key passphrase to decrypt the document.

When you decrypt the 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.

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 OpenOffice Writer, or Document Viewer. You might also need to right-click on a file and choose *Rename...* to rename a document with a proper file extension (for example, “.jpg” instead of “.jpeg”).

2.7. Decrypting on the Secure Viewing Station
2.8 Working with Documents

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

2.8.1 Handling File Formats

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

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 navigator, there will be another layer of encryption.

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:

```
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.

2.8.2 Researching Submissions

Journalists should take care to research submissions using the Tor Browser, ideally in a new Tails session for highly sensitive submissions. For more information, visit the Tails guide to working with sensitive documents.

2.8.3 Removing Metadata

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

Tails also 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 in the top left corner, Accessories, Metadata Anonymisation Toolkit.
We recommend always doing as much work as possible inside of Tails before copying documents back to your Journalist 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.

### 2.8.4 Risks From Malware

As long as you are using the latest version of Tails, you should be able to open submitted documents with a low risk of malicious files compromising the Secure Viewing Station. However, even if a compromise does occur, Tails is designed so that the next time you reboot, the malware will be gone.

Never scan QR codes from the Secure Viewing Station using a network connected device. These QR codes can contain links that your connected device will automatically visit. In general, you should take care 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 your digital security staff or Freedom of the Press Foundation for assistance.

### 2.9 Encrypting and Moving Documents to the Journalist Workstation

Before moving documents back to the Transfer Device to copy them to your workstation, encrypt them to your personal GPG key that you imported when setting up the Secure Viewing Station.

To do this, right-click on the document you want to encrypt and choose **Encrypt**...
Then choose your public key (and, if you choose, any additional keys, such as an editor’s) and click **OK**.

When you are done encrypting, you will have another document with the same filename but ending in “.gpg”. This file is encrypted to the GPG keys you selected. You can safely copy these encrypted files to the *Transfer Device* to transfer them to your workstation.
2.10 Decrypting and Preparing to Publish

Plug the Transfer Device into your workstation computer and copy over the encrypted documents. Decrypt them with GPG.

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.
Note: SecureDrop wants your feedback! Confused by something in our documentation? Let us know via an issue on GitHub or the community forum.

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, 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 add the Tails Administrator password, 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:

Logged on as securely daycare overspend handshake mossy critter cohesive egotism | Admin | Log Out

Admin Interface

Username
securely daycare overspend handshake mossy critter cohesive egotism

EditDelete

Created 0 seconds ago
Last login 0 seconds ago

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 to generate two-factor authentication tokens because it is Free Software. However, if it does not work for you for any reason, alternatives exist:

- FreeOTP for Android and for iOS (Free Software)
- 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**

Refer to the Useful Logs documentation to see the locations of files that contain relevant information while debugging issues on your SecureDrop servers.

**Note:** You can also use the securedrop-admin tool to extract logs to send to Freedom of the Press Foundation for analysis:

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

This command will produce encrypted tarballs containing logs from each server.

**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
```

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 x 450px. Simply click the **Update Instance Config** button:
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.

3.2. Common Tasks
install command again. If you see the same issue consistently, then you will need to troubleshoot it.

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)
- German (de_DE)
- Greek (el)
- Spanish (es_ES)
- French (fr_FR)
- Hindi (hi)
- Italian (it_IT)
- Norwegian (nb_NO)
- Dutch (nl)
- Portuguese, Brasil (pt_BR)
- Russian (ru)
- Swedish (sv)
- Turkish (tr)
- Chinese, Traditional (zh_Hant)
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*. After *upgrading to SecureDrop 0.7.x*, this is done by simply 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 the HidServAuth values are configured:** From `~/Persistent/securedrop.run ./securedrop-admin tailsconfig`. This will ensure your local Tails environment is configured properly.

**Note:** If you get an error during the Tails configuration step, as an Administrator, you should ensure you have four files *app-ssh-aths*, *mon-ssh-aths*, *app-journalist-aths* and *app-source-ths* in `~/Persistent/securedrop/install_files/ansible-base/`. These are used by the Tails configuration scripts to configure Tor.
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-rerunning `./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.
Passphrase Best Practices

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 Hidden 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.
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**: An air-gapped laptop running the Tails operating system from a USB stick that journalists use to decrypt and view submitted documents.

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

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

In addition to these dedicated computers, the journalist will also use their normal workstation computer:

- **Journalist Workstation**: The every-day laptop that the journalist uses for their work. The journalist will use this computer to connect to the Application Server 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, it is recommended that journalists always use the Tails operating system on their Journalist Workstation when connecting to the Application Server. Alternatively, this can also be its own dedicated computer.

These computers should all physically be in your organization’s office.
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 and connects to the Application and Monitor Servers over authenticated Tor Hidden 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 *Terminology Guide* 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 a machine that is kept offline and only ever used together with the Tails operating system on the offline USB stick. This machine will be used to generate GPG keys for all journalists with access to SecureDrop, as well as to decrypt and view submitted documents.
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.

This is also referred to as the “air-gapped computer,” meaning there is a gap between it and a computer connected to the Internet.

### 6.11 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 to generate two-factor authentication tokens because it is Free Software. However, if it does not work for you for any reason, alternatives exist:

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

### 6.12 Transfer Device

The **Transfer Device** is the physical media used to transfer encrypted documents from the **Journalist Workstation** to the **Secure Viewing Station**. Examples: a dedicated USB stick, CD-R, DVD-R, or SD card.

If you use a USB stick for the **Transfer Device**, we recommend using a small one (4GB or less). It will be necessary to securely wipe the entire device at times, and this process takes longer for larger devices.

Depending on your threat model, you may wish to only use one-time-use media (such as CD-R or DVD-R) for transferring files to and from the **SVS**. While doing so is cumbersome, it reduces the risk of malware (that could be run simply by opening a malicious submission) exfiltrating sensitive data, such as the private key used to decrypt submissions or the content of decrypted submissions.

When we use the phrase “sneakernet” we mean physically moving documents with the **Transfer Device** from one computer to another.
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. We have created a template passphrase database that you can use to get started. For more information, see the [Tails Guide](#).

**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 passphrases:

- A passphrase for the persistent volume on the Admin Live USB.
- A master passphrase for the KeePassX password manager, which unlocks passphrases to:
  - The *Application Server* and the *Monitor Server* (required to be the same).
  - The network firewall.
  - 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 key.
  - The credentials for the email account that OSSEC will send alerts to.
  - The Hidden Services values required to connect to the App and *Monitor Server*. 
The admin will also need to have a way to generate two-factor authentication tokens.

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

- FreeOTP for Android and for iOS (Free Software)
- 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’s personal GPG key.

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.
CHAPTER 8

Hardware

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.
- Dedicated physical computers for the Admin, Journalist, and Secure Viewing Station that can boot to Tails. At minimum this is 2 computers.
- 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 drive, and 1 drive for each admin and journalist.

In the sections that follow, we provide additional details on each item.

8.2 Required Hardware

8.2.1 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 machines 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 compromise 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 information, such as the plaintext of submissions or the encryption keys used to identify and access the Tor Hidden 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.

### 8.2.2 Workstations

**Note:** SecureDrop depends on the Tails operating system for its bootable USB drives. Since the release of Tails 3.0, 32-bit computers are no longer supported.

To see if you have a 64-bit machine, run `uname -m` from a terminal. If you see `x86_64`, then Tails should work on your current machine. If, on the other hand, you see `i686`, your current machine will not work with Tails 3.0 or greater. For more details, see the Tails website.

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.

**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 two additional USBs to use as bootable backups of the *SVS* and *Admin Workstation*.

**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](https://play.google.com/store/apps/details?id=org.freertp) and [for iOS](https://apps.apple.com/us/app/freeotp-totp-otp/id1055950963)).
- A dedicated hardware dongle that computes HOTP codes (e.g. a [YubiKey](https://www.yubico.com/products/yubikey)).

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

- authenticator [for the desktop](https://github.com/gnunn/authenticator) (Free Software)

**Transfer Device(s):** You need a mechanism to transfer encrypted submissions from the *Journalist Workstation* to the SVS to decrypt and view them. The most common transfer devices are DVD/CD-R discs and USB drives.

From a security perspective, it is preferable to use write-once media such as DVD/CD-R discs because it eliminates the risk of exfiltration by malware that persists on the Transfer Device (e.g. BadUSB).

On the other hand, using write-once media to transfer data is typically inconvenient and time-consuming. You should consider your threat model and choose your transfer device accordingly.

**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.

**Note:** If you cannot afford to purchase new hardware for your SecureDrop instance, we encourage you to consider re-purposing 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.

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 Optional Hardware

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

8.3.1 Offline Printer

It is often useful to print submissions from the Secure Viewing Station for review and annotation.

**Warning:** 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.

8.3.2 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 encrypted and used to store submissions that have been transferred from the Application Server to the SVS.

8.3.3 Backup Storage

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

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

8.3.4 Network Switch

If your firewall has fewer than four NICs, you will need an additional Ethernet switch to perform installation and maintenance tasks with the Admin Workstation. This switch is generally useful because it allows you to connect the Admin Workstation to your firewall’s LAN port without taking down either of the SecureDrop servers.

8.3.5 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
- Monitor Server Label
- Admin TAILS USB Drive Label
- Journalist TAILS USB Drive Label
- Secure Viewing Station TAILS USB Drive Label
- File Transfer USB Drive Label

### 8.4 Specific Hardware Recommendations

#### 8.4.1 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 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. We recommend the NUC5i5MYHE because it has a mid-range CPU (the 5th generation Intel i5), a Mini DisplayPort port for a monitor, and two USB 3.0 ports for faster OS installation and data transfer.

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.

The NUCs 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 240 GB SSDs (2.5”)
- 1x memory kit of 2x4GB sticks - You can put one 4GB memory stick in each of the servers.
Note: The D54250WYK we previously recommended has now entered End of Life and End of Interactive Support statuses. If you’re currently using this model for your SecureDrop setup, and need hardware support, you’ll need to consult the support community forum.

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.

Later NUC revisions (the NUC7 and NUC8 series) typically include onboard WiFi and Bluetooth, and may use an Ethernet chipset not supported by the default Ubuntu 14.04.5 kernel. We are investigating workarounds for both issues. If you are having trouble sourcing the NUC5i5MYHE, please contact us for more information on how to safely configure and use more recent NUCs.

Mac Minis

Caution: We have previously recommended the 2014 Apple Mac Minis (part number MGEM2) for installing SecureDrop. The 2018 Apple Mac Mini (part number MRTR2 or MRTT2) is not a viable candidate for installing SecureDrop, due to hardware support issues. The instructions below apply if you want to (re-)install SecureDrop on the 2014 version.

The 2014 Apple 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:

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

11. Type `exit`.

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

### 8.4.2 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.

### 8.4.3 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 22×30 slot and wireless antenna pre-assembled (for wireless card support)*, as advertised by the *NUC5i5MYHE* that we recommend.

### 8.4.4 Tails USBs

**Note:** Tails no longer supports 32-bit computers. Please see the note in the *Workstations* section for more details.
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.

### 8.4.5 Transfer Device

If you are using USBs for the transfer device, the same general recommendations for the Tails USBs also apply. 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).

If you are using DVD/CD-R’s for the transfer device, you will need *two* DVD/CD writers: one for burning DVDs from the *Journalist Workstation*, and one for reading the burned DVDs on the SVS. We recommend using two separate drives instead of sharing the same drive to avoid the potential risk of malware exfiltrating data by compromising the drive’s firmware. We’ve found the DVD/CD writers from Samsung and LG to work reasonably well, you can find some examples [here](#).

Finally, you will need a stack of blank DVD/CD-R’s, which you can buy anywhere.

### 8.4.6 Network Firewall

We recommend the *pfSense SG-3100*.

### 8.4.7 Network Switch

This is optional, for people who are using a firewall with less than 4 ports (the recommended firewall has 4 ports). Any old switch with more than 3 ports will do, such as the 5-port *Netgear ProSafe Ethernet Switch*. The SG-3100 sells with an internal switch on the LAN interface.

### 8.4.8 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.

### 8.4.9 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 terminology 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. 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 with two options. Select Yes to enable the persistent volume and enter your password, but do NOT click Login yet. Under ‘More Options,’ select Yes and click Forward.

Enter an Administration password for use with this specific Tails session and click Login.

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

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:
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. Be sure to enable admin privileges before you do this. In Tails 3.x, you enable admin privileges by clicking the + button under Additional Settings, then navigating to Administration Password. Enter an administration password and then click Start Tails.

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.

11.2. Correct the System Time
Journalists copy submissions from their *Journalist Workstation* to the *Secure Viewing Station* using the *Transfer Device* which can be a DVD or a USB drive.

### 12.1 Select DVDs or USBs

Using DVDs as the *Transfer Device* provides some protection against certain kinds of esoteric USB-based attacks on the *Secure Viewing Station*, but requires that you have blank DVDs, a dedicated DVD drive for the *Secure Viewing Station*, DVD drives for use with *Journalist Workstations*, and a shredder capable of destroying DVDs. Unless you are certain that you need to use DVDs as the *Transfer Device*, you should use USB drives instead. If you have chosen to use DVDs instead, there is nothing to set up now — just make sure that you have all the hardware on hand.

### 12.2 USB *Transfer Device* Configuration

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*.

#### 12.2.1 Create a USB *Transfer Device*

**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:

---

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.
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:
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:
You won’t need to memorize this passphrase or type it more than a few times, so feel free to make a good long one. Then 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.

12.2.2 Set Up the USB Transfer Device on Each Workstation

On each Journalist Workstation and Secure Viewing Station you’d like to use the Transfer Device you will securely save the passphrase on the persistent volume. This ensures that you will only have to type in the passphrase once during initial set up and it will be automatic thereafter:

1. Insert the Transfer Device
2. Go to Places Computer and click the USB drive in the left column.
3. Type in the passphrase and click “Remember Password”: 
Remember to first do this on the *Secure Viewing Station* you just used to create the device!

After you’ve done this on each computer you wish to use the *Transfer Device* with, you’re good to go!
Generate the SecureDrop Submission Key

When a document or message is submitted to SecureDrop by a source, it is automatically encrypted with the SecureDrop 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 SecureDrop Submission Key.

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 0.12.0~rc1

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:

```
$ 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) DSA (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:  

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 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 Public Key
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

**Caution:** Do not skip this step as this steps validates the files in your Git repository.

First, download and verify the SecureDrop Release Signing Key.

```
gpg --recv-key "2224 5C81 E3BA EB41 38B3 6061 310F 5612 00F4 AD77"
```

**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.

**Tip:** 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. The default GPG configuration on Tails uses a keyserver pool, which may occasionally return a malfunctioning keyserver, causing the `--recv-key` command to fail.

If the command is consistently failing after a few tries, it could indicate that the default GPG key servers are down or unreachable. As a workaround, another keyserver can be specified by adding the `--keyserver` option to the `gpg --recv-key` command. In our experience, the SKS HKPS keyserver pool is usually a reliable alternative, so try:

```
gpg --keyserver hkps://hkps.pool.sks-keyservers.net --recv-key "2224 5C81 E3BA EB41 38B3 6061 310F 5612 00F4 AD77"
```

Again, this is a keyserver pool, so you may need to retry the command a couple of times before it succeeds.

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 that the current release tag was signed with the release signing key:

```
cd ~/Persistent/securedrop/
git checkout 0.12.0-rc1
git tag -v 0.12.0-rc1
```
You should see Good signature from "SecureDrop Release Signing Key" in the output of that last command along with the fingerprint above.

**Caution:** If you do not, 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 admin 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 Section 1.3.

- 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.

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

Anti-lockout

- **Disable webConfiGurator anti-lockout rule**
  
  When this is unchecked, access to the webConfiGurator on the LAN interface is always permitted, regardless of the user-defined firewall rule set. Check this box to disable this automatically added rule, so access to the webConfiGurator is controlled by the user-defined firewall rules (ensure a firewall rule is in place that allows access, to avoid being locked out!) Hint: the "Set interface(s) IP address" option in the console menu resets this setting as well.

DNS Rebind Check

- **Disable DNS Rebinding Checks**
  
  When this is unchecked, the system is protected against DNS Rebinding attacks. This blocks private IP responses from the configured DNS servers. Check this box to disable this protection if it interferes with webConfiGurator access or name resolution in the environment.

Alternate Hostnames

- **192.168.1.1, 10.0.0.1**
  
  Alternate Hostnames for DNS Rebinding and HTTP_REFERER Checks. Specify alternate hostnames by which the router may be queried, to bypass the DNS Rebinding Attack checks. Separate hostnames with spaces.

Browser HTTP_REFERER enforcement

- **Enable HTTP_REFERER enforcement check**
  
  When this is unchecked, access to the webConfiGurator is protected against HTTP_REFERER redirection attempts. Check this box to disable this protection if it interferes with webConfiGurator access in certain corner cases such as using external scripts to interact with this system. More information on HTTP_REFERER is available from Wikipedia.
3 NIC Example (SG-3100)

![Image of setup wizard](https://example.com/image.png)

**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 Worksta-
tion 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 Con-
nection 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**: 10.20.1.1
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 screenshots 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:
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

---

15.4. SecureDrop Configuration 123
• 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”: 

---

15.4. SecureDrop Configuration
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`
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 with `--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:
SecureDrop Documentation, Release 0.12.0~rc1

System / Update / System Update

Please wait while the system update completes.
This may take several minutes. Do not leave or refresh the page!

Updating System
[3/43] Fetching php56-xmireader-5.6.31.txz: ... done
[4/43] Fetching php56-xml-5.6.31.txz: ... done
[5/43] Fetching php56-tokenizer-5.6.31.txz: ... done
[6/43] Fetching php56-sysvshm-5.6.31.txz: ... done
[7/43] Fetching php56-sysvsem-5.6.31.txz: ... done
[8/43] Fetching php56-sysvmsg-5.6.31.txz: ... done
[9/43] Fetching php56-sqlite3-5.6.31.txz: ... done

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.
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.

### 16.1 Install Ubuntu

**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 14.04.5 LTS (Trusty Tahr)**. 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-14.04.5-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.

Alternatively, you can use the command line:
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-14.04.5-server-amd64.iso SHA256SUMS)
```

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

```
ubuntu-14.04.5-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 Tails 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:

```
sudo dd conv=fdatasync if=ubuntu-14.04.5-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 the recommended 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, only choose OpenSSH server by hitting 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
- The IP address of the Monitor Server
The non-root user’s name and passphrase for the servers.

### 16.2 Test Connectivity

Now that both the network firewall and the servers are connected and configured, 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:

```bash
$ 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:

```bash
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:

```bash
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).

```bash
$ 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.
17.1 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:

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

The package installation will take approximately 10 minutes, 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*:

![Terminal](image)

**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.
17.2 Localization of the Source Interface and Journalist Interface

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

- Arabic (ar)
- German (de_DE)
- Greek (el)
- Spanish (es_ES)
- French (fr_FR)
- Hindi (hi)
- Italian (it_IT)
- Norwegian (nb_NO)
- Dutch (nl)
- Portuguese, Brasil (pt_BR)
- 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 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 SecureDrop Submission Key (from the Transfer Device)
- the SecureDrop 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 SecureDrop Submission Key public key file
- the admin’s GPG public key file (for encrypting OSSEC alerts)

The SecureDrop Submission 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`.

When you’re done, save the file and quit the editor.

### 17.4 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 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 Tor Hidden Service will be available in the following files under `install_files/ansible-base`:

- `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 hidden 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.

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.
18.1 Auto-connect to the Authenticated Tor Hidden 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 Hidden Services (ATHS) to provide authenticated access instead. Authenticated Tor Hidden 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 hidden services, restart Tails and make sure to enable persistence.
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 su
cd /var/www/securedrop
./manage.py add-admin
```

Follow the prompts.

A secure diceware 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: You can find the Onion URLs for SSH in `app-ssh-aths` and `mon-ssh-aths` inside the `install_files/ansible-base` directory.

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.
   - Do this by opening the Tor Browser and navigating to the onion URL from `app-source-ths`. Proceed through the codename generation (copy this down somewhere) and you can submit a message or attach any random unimportant 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 Tor Browser and navigate to the onion URL from `app-journalist-aths`. 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 `app-source-ths` URL 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. We highly recommend that you create persistent bookmarks for the Source Interface and Journalist Interface addresses within Tor Browser.

6. 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 Tor Hidden Service (ATHS). 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 *SecureDrop 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.

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 ATHS, we need to set up the Journalist Workstation to auto-configure Tor just as we did with 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.

Tip: Copy the files app-journalist-aths and app-source-ths from the Admin Workstation via the Transfer Device. Place these files in ~/.Persistent/securedrop/install_files/ansible-base on the Journalist Workstation, and the ./securedrop-admin tailsconfig tool will automatically use them. Don’t forget to securely delete these files from the Transfer Device when you’re done, by right-clicking them in the file manager and selecting Wipe.

Warning: Do not copy the files app-ssh-aths and mon-ssh-aths to the Journalist Workstation. Those files grant access via SSH, and only the Admin Workstation should have shell access to the servers.

Warning: The app-journalist-aths file contains a password for the authenticated hidden service used by the Journalist Interface, and should not be shared except through the onboarding process.

Since you need will the Tails setup scripts (securedrop/tails_files) that you used to Configure the *Admin Workstation* Post-Install, clone (and verify) the SecureDrop repository on the Journalist Workstation, just like you did for the Admin Workstation. Refer to the docs for cloning the SecureDrop repository, then return here to continue setting up the Journalist Workstation.

Once you’ve done this, run the install script to configure the shortcuts for the Source and Journalist Interfaces:

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

If you did not copy over the app-source-ths and app-journalist-aths files from the Admin Workstation, the script will prompt for the information. Make sure to type the information carefully, as any typos will break access for the Journalist Workstation.

Once the script is finished, you should be able to access the Journalist Interface. Open the Tor Browser and navigate to the .onion address for the Journalist Interface. You should be able to connect, and will be automatically taken to a login page.
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.

21.4.1 Import GPG Keys for Journalists with Access to SecureDrop to the SVS

While working on a story, journalists may need to transfer some documents or notes from the Secure Viewing Station to the journalist’s work computer on the corporate network. To do this, the journalist should re-encrypt them with their own keys. If a journalist does not already have a personal GPG key, they can follow the same steps above to create one. The journalist should store the private key somewhere safe; the public key should be stored on the Secure Viewing Station.

If the journalist does have a key, transfer their public key from wherever it is located to the Secure Viewing Station, using the Transfer Device. Open the file manager and double-click on the public key to import it.

21.5 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 Tails USB that they have verified they are able to boot on the Journalist Workstation.
2. Verify they are able to decrypt the persistent volume on the Journalist Tails USB.

3. Ensure that they can connect to and login to the Journalist Interface.

4. Ensure that they have a Data Transfer Device with a saved passphrase.

5. Verify they have access to the Secure Viewing Station they will be using by plugging in the SVS USB, booting, and verifying they can decrypt the persistent volume.

**Note:** Again, it is important that they test on the same SVS Tails USB and the same Secure Viewing Station they will be using on a day to day basis.

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.

At this point, the journalist has verified they have the devices and credentials they need and can proceed to a walk-through 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 hidden 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 hidden 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:

```bash
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 hidden 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:

```
Header set Cache-Control "max-age=0, no-cache, no-store, must-revalidate"
Header edit Set-Cookie ^(.*)$ $;HttpOnly
Header set Pragma "no-cache"
Header set Expires "-1"
Header always append X-Frame-Options: DENY
Header set X-XSS-Protection: "1; mode=block"
Header set X-Content-Type-Options: nosniff
Header set X-Download-Options: noopen
Header set X-Permitted-Cross-Domain-Policies: master-only
Header set Content-Security-Policy: "default-src 'self'"
Header set Referrer-Policy "no-referrer"
```

If you intend to run nginx as your webserver instead, this will work:

```
add_header Cache-Control "max-age=0, no-cache, no-store, must-revalidate";
add_header Pragma no-cache;
add_header Expires -1;
add_header X-Frame-Options DENY;
add_header X-XSS-Protection "1; mode=block";
add_header X-Content-Type-Options nosniff;
add_header X-Download-Options: noopen;
add_header X-Permitted-Cross-Domain-Policies master-only;
add_header Content-Security-Policy "default-src 'self'";
add_header Referrer-Policy "no-referrer";
```
### 23.9 Additional Apache Configuration

To enforce HTTPS/SSL always, you need to set up redirection within the HTTP (port 80) virtual host:

```apache
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
```

The same thing can be achieved in nginx with a single line:

```nginx
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:

```bash
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:

```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:

```nginx
add_header Strict-Transport-Security max-age=16070400;
ssl_protocols TLSv1.2 TLSv1.3;
ssl_prefer_server_ciphers on;
ssl_ciphers "TLS-CHACHA20-POLY1305-SHA256:TLS-AES-256-GCM-SHA384:TLS-AES-128-GCM-
...SHA256:EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH";
```

**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 Cipherli.st.

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.

**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.

**Don’t log access to the *Landing Page* in the webserver**

Here’s an Apache example that would exclude the *Landing Page* from logging:

```apache
SetEnvIf Request_URI "^/securedrop$" dontlog
CustomLog logs/access_log common env=!dontlog
```
In nginx, logging can be disabled like so:

```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.
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 hidden 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](http://torproject.org) to download it.
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 what type of 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].
Pre-Install Hardware Checklist

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.
- Dedicated physical computers for the Admin, Journalist, and Secure Viewing Station that can boot to Tails. At *minimum* this is 2 computers.
- 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 drive, and 1 drive for each admin and journalist.

**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.
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 home page, 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:
What Makes SecureDrop Unique

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.
CHAPTER 30

Useful Logs

For tips on collecting these logs, see the admin guide.

30.1 Both servers

- AppArmor and grsec errors: /var/log/kern.log
- iptables: /var/log/syslog

30.2 Application Server

- Apache: /var/log/apache2/*

If an error is triggered it’s in the SecureDrop application logs: /var/log/apache2/source-error.log and /var/log/apache2/journalist-error.log

30.3 Monitor Server

- OSSEC

  /var/ossec/logs/ossec.log
  /var/ossec/logs/alerts/alerts.log

- Postfix/Procmail

  /var/log/mail.log
  /var/log/procmail.log
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.
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`:

```
smtplib_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
SecureDrop Documentation, Release 0.12.0-rc1

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 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:
### Problem | Solution
--- | ---
Connection timed out | Check that the hostname and port is correct in the relayhost line of `/etc/postfix/main.cf`
Server certificate not verified | 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.
Authentication failure | 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.

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:
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 -connect 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 GeoTrust Global CA. 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 \texttt{-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.

\begin{verbatim}
Received From: (app)
Rule: 2902 fired (level 7) -> "New (Debian Package) installed."
Portion of the log(s):
status installed <package name> <version>
\end{verbatim}

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.

\begin{verbatim}
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.
\end{verbatim}

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 securedrop@freedom.press for help.

31.3.2 Uncommon OSSEC Alerts

If you believe that the system is behaving abnormally, you should contact us at the SecureDrop Support Portal or securedrop@freedom.press for help.
To log-in to SecureDrop and retrieve messages sent by sources, the journalist must be using the Tails operating system. The admin must also use Tails to access the Journalist Interface and create new users.

If you followed the SecureDrop Installation instructions correctly, you should have already created a Journalist Workstation Tails USB and an Admin Workstation Tails USB and enabled the persistence volume on each. If you have not, or need to create another Tails USB for a second journalist, follow the steps below.

If you already know how to boot the Admin Workstation or Journalist Workstation Tails USB with persistence, you can skip down to the step ‘download the repository’.

Note that for all of these instructions to work, you should have already installed the main SecureDrop application. It is also required that you use Tails version 2.x or greater.

### 32.1 Installing Tails on USB Sticks

Tails is a live operating system that is run from removable media, such as a DVD or a USB stick. For SecureDrop, you’ll need to install Tails onto USB sticks and enable persistent storage.

We recommend creating an initial Tails Live USB or DVD, and then using that to create additional Tails Live USBs with the Tails Installer, a special program that is only available from inside Tails. You will only be able to create persistent volumes on USB sticks that had Tails installed 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 Tails USB stick. Here are some links to help you out:

- Download and verify the Tails .iso
- Install onto a USB stick or SD card
- Create & configure the persistent volume
32.1.1 Note for macOS Users Manually Installing Tails

The Tails documentation for manually installing Tails onto a USB device on macOS describes how to copy the downloaded .iso image to a USB stick in Section 4, “Do the copy”. This section includes the following `dd` invocation to copy the .iso to the USB:

```
dd if=[tails.iso] of=/dev/diskX
```

This command is very slow. In our testing, it took about 18 minutes to copy the .iso to a USB 2.0 drive. You can speed it up by changing the arguments to `dd` like so:

```
dd if=[tails.iso] of=/dev/rdiskX bs=1m
```

Note the change from `diskX` to `rdiskX`. This reduced the copy time to 3 minutes for us.

32.2 Configure Tails for Use with SecureDrop

32.2.1 Persistence

Creating an encrypted persistent volume will allow you to securely save information in the free space that is left on the Transfer Device. This information will remain available to you even if you reboot Tails. Instructions on how to create and use this volume can be found on the Tails website. You will be asked to select from a list of persistence features, such as personal data. We require that you enable all features.

32.2.2 Start Tails and Enable the Persistent Volume

When starting Tails, you should see a “Welcome to Tails” screen with two options. Select Yes to enable the persistent volume and enter your passphrase. Select Yes to show more options and click Forward. Enter an Administration passphrase for use with this specific Tails session and click Login.

32.2.3 Download the Repository

The rest of the SecureDrop-specific configuration is assisted by files stored in the SecureDrop git repository. To get started, open a terminal and run the following commands to download the git repository. Note that since the repository is fairly large and Tor can be slow, this may take a few minutes.

```
cd ~/Persistent
git clone https://github.com/freedomofpress/securedrop.git
```

32.2.4 Passphrase Database

We provide a KeePassX passphrase 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 passphrase 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 passphrase, 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 passphrases if you forget the master passphrase or the location of the key file used to protect the database.

32.2.5 Set Up Easy Access to the Journalist Interface

To complete setup of the Admin Workstation or Journalist Workstation, we recommend using the scripts in tails_files to easily configure Tor to access the Journalist Interface.

Navigate to the directory with the setup scripts and begin the installation by typing these commands into the terminal:

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

Type the administration passphrase that you selected when starting Tails and hit enter. This installation script does the following:

• Downloads additional software

• Installs a program that automatically and persistently configures Tor to access the SecureDrop servers and interfaces, by adding HidServAuth values to /etc/tor/torrc

• Sets up desktop and main menu shortcuts for the Journalist Interface and Source Interface

• Sets up SSH host aliases for mon and app

• Makes it so that Tails installs Ansible at the beginning of every session

If you are missing any files, the script will exit with an error. If you’re running this script as an admin, the entire setup should be automatic.

If you’re running the script as a journalist, you will need the .onion addresses for each interface, provided to you by the admin.

We use an “authenticated” Tor Hidden Service so that adversaries cannot access the Journalist Interface, providing a layer of defense-in-depth which protects the Journalist Interface even if there is a security vulnerability in the web application, or if the journalist’s username, passphrase, and two-factor token are stolen. The extra configuration that is required is handled by this script.

Our ./securedrop-admin tailsconfig tool sets up Tails to work with SecureDrop every time you login. As long as Tails is booted with the persistent volume enabled then you can open the Tor Browser and connect to the Journalist Interface as normal.
32.2.6 Create Bookmarks for Source Interface and Journalist Interface

If you want, you can open the browser and create bookmarks for the Source and Journalist Interfaces. Navigate to the site you wish to bookmark, select Bookmarks and Bookmark This Page, give the site a useful name (e.g. Source Interface), and click Done. Tails will remember the bookmarks even if you reboot.
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 most 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, and provides some troubleshooting tips that you may find useful if you are trying to use a different printer.

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 along with enabling persistence; 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.
33.1 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 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:

![Printers](image2)

This indicates the 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.
33.2 Troubleshooting

For instances where your printer does not work out of the box, most difficulties stem from not selecting the right driver (extra software needed for the printer and computer to communicate). Luckily, Tails has a large number of drivers for just about any popularly manufactured printer on hand, without even having to download new drivers from the web.

Just as before, if you’ve enabled persistence, boot with your persistent volume, and set an admin passphrase.

Make sure your computer is NOT connected to the Internet. This will make sure that your printer set-up is never influenced by a network connection.

Plug in your printer and navigate to Printing. Applications System Tools Settings Printing.
Click Add.

Immediately, Tails will recognize the plugged-in printer, and make the best suggestion from its on-board database of printer drivers.
Tails will guide you through a default set-up, suggesting the best match for the printer you have. These choices come from Tails’ pre-installed driver database.
The recommended driver does not always match the actual make and model of your printer, but starting with the recommendations is a good idea. Sometimes you get lucky, and Tails suggests a perfect match. Click **Forward**, and **Apply** your settings.

You’ll notice that the printer is now listed in your Printing Configurations in your persistent storage.
The only way to be sure you have the right driver is by doing a test print. Right-click on your new printer config and select **Properties** to open its settings, then click **Print Test Page**.
In this initial test, the recommended driver was wrong! My test page came out garbled, and my printer gave me a warning that I had to manually clear before the page printed.
Don’t worry if this happens to you; you can edit the printer configuration to point it to the correct driver for your model. Select **Properties** again and choose **Change...** next to the “Make and Model” directive.
To fix this problem, I selected the CUPS + Gutenprint driver, even though it wasn’t recommended. Click **Forward** to save your changes.

Do another test print, checking your printer for indicators that it’s working or not. This time, printing works perfectly. If you still experience garbled text, try another driver from your selections. It is a process of trial-and-error.
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:
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.
HTTPS on the Source Interface

The SecureDrop Source Interface is served over a Tor Hidden Service, requiring a *.onion URL to access it. While Tor Hidden 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.

- The cryptographic primitives used by Tor Hidden Services are considered to be outdated, and while there are no known compromises of the security of Tor Hidden Services due to this issue, you may wish to provide an additional layer of transport encryption using stronger cryptographic primitives, which is most easily achieved by setting up HTTPS on the Source Interface.

Note: This issue is being addressed by the Tor Project with their Next Generation Onion Services design, but the implementation of the new design is still a work in progress and is not expected to be deployed until December 2017 at the earliest.

34.1 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.

### 34.2 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.
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.

### 35.1 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 not 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
```
This is a high level schedule for what happens for the 2 days during an on-site install.

### 36.1 Day 1: Preparation and Install

#### 36.1.1 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

#### 36.1.2 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

36.1.3 Questions

Time: 30 min

36.1.4 Installing SecureDrop

Time: 6 hours
• Follow Installing SecureDrop

36.2 Day 2: Journalist and Admin Training

36.2.1 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

### 36.2.2 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 hidden 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
CHAPTER 37

Using a YubiKey with the Journalist Interface

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.

### 37.1 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.

### 37.2 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
   
   ```
   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
   
   ```
   yubikey-personalization-gui
   ```
37.3 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.

### 37.4 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.

### 37.5 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.

38.1 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 srm to securely erase files, which takes significantly more time than normal file deletion. You can monitor the progress of queued deletion jobs with sudo tail -f /var/log/securedrop_worker/err.log.
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.

### 38.2 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.

#### 38.2.1 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 HidServAuth values from `install_files/ansible-base/app-ssh-aths` and `install_files/ansible-base/mon-ssh-aths` are in Tails at `/etc/tor/torrc`

(If Ansible is not installed, or the HidServAuth values are missing or incorrect, see *Configure the Admin Workstation Post-Install* for detailed instructions.)

#### 38.2.2 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.
38.3 Restoring

38.3.1 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.

38.3.2 Restoring From a Backup File

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:

- app-source-ths
- app-journalist-aths
- app-ssh-aths

Then rerun ./securedrop-admin tailsconfig to update the Admin Workstation to use the restored Onion URLs again. See Configure the Admin Workstation Post-Install for detailed instructions.

38.4 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.
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.

### 39.1 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.

39.2 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.
320 GB Hard Disk
WDC WD3200BEKT-75PVMT1

CD/DVD Drive
MATSUSHITA DVD- RAM U88A0

16 GB Drive
General USB Flash Disk

16 GB Drive
General USB Flash Disk

1.1 GB Loop Device
/dev/mmcblk/mnt/mnt/var...system.squashfs

16 GB Drive
/dev/sdc

Model General USB Flash Disk (1.00)
Size 16 GB (15,640,600,576 bytes)
Partition Master Boot Record
Serial Number 04744D0000009061

Volumes

My usb drive
Partition 1
16 GB/FAT

Size 16 GB (15,639,552,000 bytes)
Device /dev/sdc:1
Partition Type W95 FAT32 (LBA)
Contents FAT (32-bit version) — Not Mounted

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.
Open a Nautilus window with administrative privileges by first going to **Applications System Tools Root Terminal.**
Type `nautilus` at the terminal prompt and hit enter. You’ll need to type in your administrative passphrase you set at the Tails login screen.
Note: When you run `nautilus`, you may run into an error where Nautilus complains that it can’t create a required folder. If that happens, just click OK and continue normally.

If you can’t find the “Root Terminal” window, it might be because an administrator passphrase wasn’t set when you logged in to Tails. If that’s the case, you’ll need to restart Tails and set one at the login screen.

**Warning:** Make sure you use keep the `Root Terminal` window open while you perform the backups. Otherwise, the `Nautilus` window will close.

Make sure you create a directory on the backup drive to store the data from the drive you are backing up:
Copy over everything in the TailsData partition to the relevant folder on the Backup drive. You can simply drag to select all the files and then copy and paste them to the relevant folder on the Backup drive.

In particular, ensure gnupg and Persistent have been successfully copied over. These files are critical for decrypting submissions.

Once complete, unmount the TailsData partition.

Repeat these steps for every device, making a new folder on the backup device for each device you backup.

Finally, once you have completed the steps described in this section for each USB drive, unmount the Backup partition and store the drive somewhere safely.
Troubleshooting Kernel Updates

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.

40.1 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.
Press the “F10” key to boot.

40.2 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.

40.3 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:

```
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.
```
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.

### 40.4 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.

### 40.5 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:

```
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.

### 40.6 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.

![GRUB Menu]

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.

### 41.1 Community Based Support

The SecureDrop forum 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 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.**

### 41.2 Priority Support and Training

Freedom of the Press Foundation provides paid priority support and SecureDrop training to organizations. Visit the Priority Support and Training pages on the SecureDrop website for more information.
On 30 April 2019, Ubuntu 14.04 LTS (Trusty) will reach End of Life. After this date, no new security updates to the base operating system will be provided. It is therefore of critical importance for the security of all SecureDrop instances to upgrade to Ubuntu 16.04 LTS (Xenial) before April 30.

SecureDrop servers provisioned before February 26 use Ubuntu 14.04 LTS as the base operating system. Support for Ubuntu 16.04 LTS (which will receive security updates until April 2021) is scheduled to be included with the next release of SecureDrop, version 0.12.0, on February 26. The operating system update itself must be performed manually.

We recommend that you plan two working days (after your instance has been updated to SecureDrop 0.12.0) to backup your instance, perform the upgrade, and test your instance once it is upgraded. We recommend scheduling this maintenance window no earlier than March 6.

Anytime before then, we suggest taking some simple preparatory steps to ensure your SecureDrop instance can be upgraded smoothly.

### 42.1 Preparation Procedure

In summary, the preparation procedure consists of:

1. ensuring your instance is running the latest version of SecureDrop;
2. ensuring your Admin Workstation and Journalist Workstations are up to date;
3. ensuring you have a recent backup of the Securedrop servers;
4. verifying that you still have SSH access to the servers.

Each of these steps are described below.
42.2 Checking your Server Securedrop version

The simplest way to check your SecureDrop version is to load the .onion address of your Source Interface in the Tor Browser. The version number will be in the footer of the Source Interface.

If you have already confirmed that you have SSH access to the servers, you can also check the application version from the command line on the Application Server by running the command:

```
apt-cache policy securedrop-app-code
```

SecureDrop servers are updated automatically with the latest release version (0.11.1 as of January 24, 2019). Recently, some long-running SecureDrop instances were affected by a bug which will cause any updates after 0.10.0 to fail. If your instance is still running 0.10.0, please consult our advisory to update to the latest version.

**Important:** If your instance is affected by this bug, it will no longer receive automatic updates. This is a major security risk and we urge you to take manual action as soon as possible to update SecureDrop. Please do not hesitate to contact us if we can help.

42.3 Ensuring your Admin Workstation is up-to-date

First, back up your Admin Workstation. using the process described here: Back up the Workstations.

Next, ensure you are running the latest version of the Tails OS. You can do this by starting up your Admin Workstation and selecting Activities > Tails > About Tails. If you are running a version prior to 3.11, you will need to upgrade to version 3.11. For more information on upgrading your Tails USB, see Upgrading a Tails USB stick.

To check the version of the SecureDrop code installed on your Admin Workstation, start the Admin Workstation with its persistent volume unlocked and an administration password set. Then open a terminal window and run the following commands:

```
cd ~/Persistent/securedrop
git status
```

The output from `git status` should include the following text:

```
HEAD detached at <version>
```

where `<version>` is the version of the workstation code that is installed.

If the Admin Workstation is at version 0.11.1, it is up-to-date, and you can proceed with making a backup of the instance and verifying SSH connectivity. If the Admin Workstation is running an earlier version, you will need to upgrade it, using the appropriate steps for your version:
• 0.9.1 to 0.10.0: *Upgrading from version 0.9.1 or later.*

• 0.4 to 0.9.0: *Upgrading from versions 0.4-0.9.0.***

• Earlier than 0.4: *Upgrading from version 0.3.x - reprovisioning your Admin Workstation.*

### 42.3.1 Upgrading from version 0.9.1 or later

If the workstation is at least at version 0.9.1 and up to version 0.10.0, you should see a graphical updater informing you about the availability of a new version. The graphical updater looks like this:
Follow the graphical prompts to complete the update. If you don’t see the graphical updater, make sure that you start up the Admin Workstation with both an Administration password set and the persistent volume unlocked.
If you still can’t see the graphical prompt, then you can update manually by following the instructions for upgrading from versions 0.4 to 0.9.0.

### 42.3.2 Upgrading from versions 0.4-0.9.0

If your workstation code version is between 0.4 and 0.9.0, then you will need to update to the latest version manually. First, open a terminal window and run the following commands:

```bash
cd ~/Persistent/securedrop
git fetch --tags
gpg --recv-key "2224 5C81 E3BA EB41 38B3 6061 310F 5612 00F4 AD77"
git tag -v 0.11.1
```

The output should include the following two lines:

```text
gpg: using RSA key 22245C81E3BAEB4138B36061310F561200F4AD77
gpg: Good signature from "SecureDrop Release Signing Key"
```

Please verify that each character of the fingerprint above matches what is on the screen of your workstation. If it does, you can check out the new release:

```bash
git checkout 0.11.1
```

**Important:** If you see the warning refname ‘0.11.1’ is ambiguous in the output, we recommend that you contact us immediately at securedrop@freedom.press (GPG encrypted).

Finally, run the following commands:

```bash
./securedrop-admin setup
./securedrop-admin tailsconfig
```

### 42.3.3 Upgrading from version 0.3.x - reprovisioning your Admin Workstation

If your Admin Workstation has not been updated since version 0.3 of SecureDrop was released, its Tails version is most likely also out-of-date. In this case, we recommend provisioning a new Admin Workstation using the configuration information from the old workstation.

First, prepare a new Tails USB stick with a persistent volume, using the latest version of Tails. For more information on this process, see [Create Tails USBs](#). This will be your new Admin Workstation.

Start up your new Admin Workstation with its persistent volume unlocked and an administration password set.

Open a terminal and run the following commands to install the SecureDrop app code:

```bash
gpg --recv-key "2224 5C81 E3BA EB41 38B3 6061 310F 5612 00F4 AD77"
cd ~/Persistent
`
Note: The SecureDrop application code must be installed in the ~/Persistent/securedrop directory in order to complete the reprovisioning process successfully. Do not install it in a different location.

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 is on the screen of your workstation. If it does, you can check out the new release. If it does not, signature verification has failed and you should not proceed with the installation. If this happens, please contact us at securedrop@freedom.press.

Now, check out the current release with the following command:

```
git checkout 0.11.1
```

Next, mount the persistent volume of the old Admin Workstation in order to retrieve instance-specific files that you’ll need to set up the new workstation. To do so:

1. Plug your old Admin Workstation into a free USB port
2. Browse to **Places > Computer** in the Tails top navigation bar.

3. Click the encrypted volume in the left-hand panel of the file browser, and enter the decryption passphrase for the volume to mount it.
Next, copy the files that you’ll need for the new Admin Workstation. Open a terminal and run the following commands:

```bash
cp /media/amnesia/TailsData/openssh-client/* ~/.ssh/
export SRC="/media/amnesia/TailsData/Persistent/securedrop/install_files/ansible_base"
export DST="~/Persistent/securedrop/install_files/ansible-base"
cp $SRC/{app,mon}* $DST/
cp $SRC/prod-specific.yml $DST/
```

# Next, you’ll need to copy over the instance’s submission public key and OSSEC public key. Their filenames may vary, but you can check them in the instance configuration file using the following command:

grep "_public_key" $DST/prod-specific.yml

# Assuming that their names are `Securedrop.asc` and `ossec.asc` respectively, you should then copy them across by running the following commands:

cp $SRC/Securedrop.asc $DST/
cp $SRC/ossec.asc $DST/
```

If you use Tails’ KeepassX password manager to store instance-specific passwords, you should also copy over the old workstation’s KeepassX database. The default location for the KeepassX database is `/media/amnesia/TailsData/Persistent/securedrop-keepassx.kdbx`. Copy it to the new Admin Workstation’s persistent volume with the following command:

```bash
cp /media/amnesia/TailsData/Persistent/securedrop-keepassx.kdbx ~/Persistent/
```
Once the instance-specific files have been copied across, unmount the old Admin Workstation’s persistent volume by clicking its Eject icon in the file browser.

Next, you’ll need to configure the new Admin Workstation using the copied files. In a terminal, run the following commands:

```bash
cd ~/Persistent/securedrop
./securedrop-admin setup
./securedrop-admin tailsconfig
```

You can now proceed to back up your instance and test SSH connectivity, as described below.

### 42.4 Backing up your instance

Once your Admin Workstation is up-to-date, you should delete any previously-downloaded submissions and sources via the Journalist Interface before backing up the instance servers. In general, this should be done by or in coordination with the editorial staff responsible for the instance!

**Important:** Deleting old submissions is good security practice. It’s also important in order to control the size of backups, as the backup files are transferred to the Admin Workstation over the Tor network.

To back up your instance servers, open a terminal on the Admin Workstation and run the following commands:

```bash
cd ~/Persistent/securedrop
./securedrop-admin setup
./securedrop-admin backup
```

Once the command is completed, you will find the backup files in the `~/Persistent/securedrop/install_files/ansible-base` directory. We recommend that you store those on an encrypted volume on a separate USB stick for safe keeping. For more information on the backup process, see *Backup, Restore, Migrate*.

### 42.5 Verifying SSH access

Check to see if you can still access the servers via SSH. To do this, start up your Admin Workstation (with persistent storage unlocked) and run the following commands.

```bash
ssh app hostname  # command output should be 'app'
ssh mon hostname  # command output should be 'mon'
```

If you are having trouble accessing the servers via SSH, try the following:

- creating a new Tor network circuit by disconnecting and reconnecting your Internet link, and repeating the check;
- running the `.securedrop-admin tailsconfig` command and repeating the check;
- verifying that the Source and Journalist Interfaces are available via their desktop shortcuts;
- verifying that the Application and Monitor servers are up;
- contacting us for assistance.
42.6 Upgrading Journalist Workstations

You should keep your Journalist Workstations in sync with the SecureDrop version in use on your Admin Workstation.

You can check the SecureDrop code versions on a Journalist Workstation using the procedure described above.

- If your Journalist Workstation code version is 0.9.1 or later, you can upgrade it using the graphical updater.
- If its code version is later than 0.4 and earlier than 0.9.1, you can use the process described above for an Admin Workstation with the same code version to upgrade it.
- If its code version is less than 0.4, we recommend provisioning a new Journalist Workstation instead, after upgrading your Admin Workstation.

42.7 Contact us

If you have questions or comments regarding the coming upgrade to Ubuntu 16.04 LTS or the preparatory procedure outlined above, please don’t hesitate to reach out:

- via our Support Portal, if you are a member (membership is approved on a case-by-case basis);
- via securedrop@freedom.press (GPG encrypted) for sensitive security issues (please use judiciously);
- via our community forums.
43.1 Updating the Tails Workstations

We recommend that you update all Tails drives to version 3.11, which was released concurrently with SecureDrop 0.11.0 on December 11, 2018. 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. 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 a graphical updater, you may be running an older version of the SecureDrop code on your workstation (earlier than 0.7.0). You can update as follows:

```
cd ~/Persistent/securedrop
git fetch --tags
gpg --recv-key "2224 5C81 E3BA EB41 38B3 6061 310F 5612 00F4 AD77"
git tag -v 0.11.0
```

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 0.11.0
```

**Important:** If you do see the warning “refname ‘0.11.0’ 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
```

**Important:** If you haven’t already updated your workstations when SecureDrop 0.9.1 was released, you’ll need to update your workstations manually. Due to a bug in the graphical SecureDrop updater that was fixed in SecureDrop 0.9.1 (released on September 6, 2018), attempting an update of your SecureDrop workstation code on your *Journalist or Admin Workstations* using the graphical updater may fail with an error message: “WARNING: Signature verification failed.”

Should you encounter this message, follow the instructions in the *Upgrade from 0.8.0 to 0.9.1* guide.

### 43.2 Troubleshooting Kernel Issues

If you have previously downgraded your kernel, as part of the the upgrade process to SecureDrop 0.11.0, the default Linux kernel will change to the latest released kernel (version 4.4.162).

We have tested this kernel extensively against *recommended hardware* and other common configurations. Please consult our *kernel troubleshooting guide* for instructions on how to compare the differences between kernel versions and how to roll back to an earlier version if necessary.

**Important:** The 3.14.x series kernel will be removed as part of the upgrade to SecureDrop release 0.11.0. Your system will boot to the latest kernel by default (version 4.4.162). If you experience issues after the upgrade, please report kernel compatibility issues as soon as possible.
43.3 Getting Support

Should you require further support with your SecureDrop installation or upgrade, we are happy to help!

- Community support is available at https://forum.securedrop.org
- The Freedom of the Press Foundation offers training and priority support services. See https://securedrop.org/priority-support/ for more information. If you are already a member of our support portal, please don’t hesitate to open a ticket there.
Upgrade from 0.9.1 to 0.10.0

44.1 Updating the Tails Workstations

We recommend that you update all Tails drives to version 3.10, which was released concurrently with SecureDrop 0.10.0 on October 23, 2018. 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. 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 a graphical updater, you may be running an older version of the SecureDrop code on your workstation (earlier than 0.7.0). You can update as follows:

```
cd ~/Persistent/securedrop
git fetch --tags
gpg --recv-key "2224 5C81 E3BA EB41 38B3 6061 310F 5612 00F4 AD77"
git tag -v 0.10.0
```

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 0.10.0
```

**Important:** If you do see the warning “refname ‘0.10.0’ 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
```

**Important:** If you haven’t already updated your workstations when SecureDrop 0.9.1 was released, you’ll need to update your workstations manually. Due to a bug in the graphical SecureDrop updater that was fixed in SecureDrop 0.9.1 (released on September 6, 2018), attempting an update of your SecureDrop workstation code on your Journalist or Admin Workstations using the graphical updater may fail with an error message: “WARNING: Signature verification failed.”

Should you encounter this message, follow the instructions in the *Upgrade from 0.8.0 to 0.9.1* guide.

### 44.2 Troubleshooting Kernel Issues

If you have previously downgraded your kernel, as part of the the upgrade process to SecureDrop 0.10.0, the default Linux kernel will change to the latest released kernel (version 4.4.144).

We have tested this kernel extensively against recommended hardware and other common configurations. Please consult our *kernel troubleshooting guide* for instructions on how to compare the differences between kernel versions and how to roll back to an earlier version if necessary.

**Important:** The 3.14.x series kernel will be removed with release 0.11.0 (scheduled for December 11). Please report kernel compatibility issues immediately to avoid extended downtime.
44.3 Getting Support

Should you require further support with your SecureDrop installation or upgrade, we are happy to help!

- Community support is available at https://forum.securedrop.org
- The Freedom of the Press Foundation offers training and priority support services. See https://securedrop.org/priority-support/ for more information. If you are already a member of our support portal, please don’t hesitate to open a ticket there.
Upgraded from 0.8.0 to 0.9.1

45.1 Updating the Tails Workstations

We recommend that you update all Tails drives to version 3.9, which was released concurrently with SecureDrop 0.9.0 on September 5, 2018. Follow the Tails graphical prompts on your workstations to perform this upgrade.

Due to a bug in the graphical SecureDrop updater that was fixed in SecureDrop 0.9.1 (released on September 6, 2018), attempting an update of your SecureDrop workstation code on your Journalist or Admin Workstations using the graphical updater will fail with an error message: “WARNING: Signature verification failed.” You need to update your workstations manually by following the steps below.

1. Start the Journalist or Admin Workstation with Persistent Storage unlocked and an administration password set.
2. Once you have a working Tor connection, open a Terminal by selecting from the menu: Applications Favorites Terminal
3. Change your working directory to the SecureDrop installation directory using the command: cd ~/Persistent/securedrop/
4. Verify the installed version using the command: git status

The command output will include the line “HEAD detached at <version>”, where “<version>” is your current installed version.

45.1.1 If the workstation code is at version 0.7.0 or earlier:

Update the workstation code to version 0.8.0 (not 0.9.0 or 0.9.1) manually before proceeding:

```
git fetch --tags
gpg --recv-key "2224 5C81 E3BA EB41 38B3 6061 310F 5612 00F4 AD77"
git tag -v 0.8.0
```

The output should include the following two lines:
Please verify that each character of the fingerprint above matches what you see on the screen of your workstation. If it does, you can check out version 0.8.0:

```
git checkout 0.8.0
```

Please verify that the output of this command does not contain the text “warning: refname ‘0.8.0’ is ambiguous”.

**Important:** If you do see the warning “refname ‘0.8.0’ is ambiguous” in the output, we recommend that you contact us immediately at securedrop@freedom.press (GPG encrypted).

Run the following command:

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

Now, proceed with the 0.8.0 to 0.9.1 update as described below.

### 45.1.2 If the workstation code is at version 0.8.0:

Complete the following steps to upgrade your workstation to version 0.9.1:

1. Update the workstation code using the command: `./securedrop-admin update`
2. Update the workstation dependencies using the command: `./securedrop-admin setup`
3. Update your workstation Tails settings using the command: `./securedrop-admin tailsconfig`

You will be prompted for the temporary administration password set on the Tails greeter screen.

This should address the problem with the graphical updater for future releases.

### 45.2 Database Migration May Take Time to Complete

Some of the changes in SecureDrop 0.9.0 require a database migration that can take additional time during the upgrade process, especially if you are not using an SSD on your Application Server, or if you store data about a large number of sources and submissions on the server. This is normal behavior during the automatic upgrade, and the **Source Interface** and **Journalist Interface** will become available again once the upgrade has been completed. In case of a service outage of more than two hours, please don’t hesitate to reach out to us.

### 45.3 New Dialogs in Tails 3.9

When you run the `securedrop-admin setup` command, Tails as of version 3.9 will prompt to install packages automatically from persistent storage on each boot. These apt packages don’t need to persist; click on **Install Only Once** when you see this dialog:
45.4 Troubleshooting Kernel Issues

SecureDrop 0.9.0 ships with an update of the Linux kernel running on your Application and Monitor Servers, from version 4.4.135 to version 4.4.144. If you have not previously changed your default kernel, your server will boot into the new kernel automatically on its next reboot.

We have tested this kernel extensively against recommended hardware and other common configurations. Please consult our kernel troubleshooting guide for instructions on how to compare the differences between kernel versions and how to roll back to an earlier version if necessary.

**Important:** It is of critical importance for the security and stability of your instance that you report kernel compatibility issues to us as soon as you become aware of them.

45.5 Enabling the New Kernel After a Downgrade

If you have previously downgraded your kernel to the 3.14.x series due to compatibility issues with the kernel that shipped with SecureDrop 0.7.0 or later, we urge you to test the latest kernel (version 4.4.144).

You can test the new kernel without downtime by following our instructions for testing and enabling a new kernel after a downgrade. Please note that this is only necessary if you have manually downgraded the kernel; otherwise, the new kernel will be enabled automatically.

**Important:** The next regular release of SecureDrop, version 0.10.0, will no longer preserve a preference for a downgraded kernel. If you have downgraded your kernel, testing the new kernel and reporting compatibility issues is of critical importance to minimize the risk of an outage of your SecureDrop instance.

45.6 Getting Support

Should you require further support with your SecureDrop installation or upgrade, we are happy to help!

- Community support is available at https://forum.securedrop.org
- The Freedom of the Press Foundation offers training and priority support services. See https://securedrop.org/priority-support/ for more information. If you are already a member of our support portal, please don’t hesitate to open a ticket there.
CHAPTER 46

Upgrade from 0.7.0 to 0.8.0

46.1 Updating the Tails Workstations

We recommend that you update all Tails drives to version 3.8, which was released concurrently with SecureDrop 0.8.0 on June 26, 2018. Follow the graphical prompts on your workstations to perform this upgrade.

For the Journalist Workstations and the Admin Workstation, the graphical SecureDrop updater will also prompt you to update the SecureDrop code on your workstations. The updater was introduced in SecureDrop 0.7.0. It looks like this:
Due to bug 3567, if this update is performed using the graphical updater, a branch containing unsigned code could take precedence over a release tag, if they have the same name. The likelihood of an exploit is low, as it would require accidental or deliberate branch creation, or compromise of GitHub and/or the communication flow between
To completely eliminate this risk, we recommend that you select “Update Later” and perform the update manually by issuing the following commands on each workstation:

```bash
cd ~/Persistent/securedrop
git fetch --tags
gpg --recv-key "2224 5C81 E3BA EB41 38B3 6061 310F 5612 00F4 AD77"
git tag -v 0.8.0
```

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 you see on the screen of your workstation. If it does, you can check out the new release:

```bash
git checkout 0.8.0
```

Please verify that the output of this command does not contain the text “warning: refname ‘0.8.0’ is ambiguous”.

**Important:** If you do see the warning “refname ‘0.8.0’ is ambiguous” in the output, we recommend that you contact us immediately at securedrop@freedom.press (GPG encrypted).

Finally, run the following command:

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

Please note that this only updates the SecureDrop code on the workstation. Tails upgrades still have to be performed separately.

## 46.2 Removal of Two-Factor Authentication for Keyboard Login

SecureDrop 0.8.0 removes the requirement for two-factor authentication when logging into your server using an attached physical keyboard. This feature provided no real security benefit, as it could easily be bypassed using single user mode.

To ensure that you can login as the admin user using a physical keyboard, you may wish to use this opportunity to cycle the admin user password on your SecureDrop servers.

To do so, log into each SecureDrop server via SSH using your Admin Workstation. Become the root user by typing `sudo su`, then change the password for the admin user by typing `passwd <username>`, e.g., if your admin user account was called alice, you would type `passwd alice`.

Enter a secure password and store it in the KeePassX password manager on your Admin Workstation.

## 46.3 Troubleshooting Kernel Issues

SecureDrop 0.8.0 ships with an update of the Linux kernel running on your Application and Monitor Servers, from version 4.4.115 to version 4.4.135. If you have not previously changed your default kernel, your server will boot into the new kernel automatically on its next reboot.
We have tested this kernel extensively against recommended hardware and other common configurations. Compared with the previous kernel, it ships with additional hardware support. If you are experiencing an outage after the update, it may be due to differences between the two kernel versions relevant to your specific configuration.

Please consult our kernel troubleshooting guide for instructions on how to compare the differences between kernel versions and how to roll back to an earlier version if necessary.

**Important:** It is of critical importance for the security and stability of your instance that you report kernel compatibility issues to us as soon as you become aware of them.

## 46.4 Enabling the New Kernel After a Downgrade

If you have previously downgraded your kernel to the 3.14.x series due to compatibility issues with the kernel that shipped with SecureDrop 0.7.0 (version 4.4.115), we urge you to test the new kernel (version 4.4.135). The new kernel ships with expanded hardware support and is intended to address the hardware compatibility issues that we are aware of.

You can test the new kernel without downtime by following our instructions for testing and enabling a new kernel after a downgrade. Please note that this is only necessary if you have manually downgraded the kernel; otherwise, the new kernel will be enabled automatically.

If the new kernel does not address compatibility issues on your hardware, please let us know as soon as you can by following our instructions for reporting compatibility issues. We intend to remove support for kernel series 3.14.x in a future release, once all major compatibility issues we are aware of have been resolved.

## 46.5 Getting Support

Should you require further support with your SecureDrop installation or upgrade, we are happy to help!

- Community support is available at https://forum.securedrop.org
- The Freedom of the Press Foundation offers training and priority support services. See https://securedrop.org/priority-support/ for more information. If you are already a member of our support portal, please don’t hesitate to open a ticket there.
47.1 Updating the Tails Workstations

All Tails drives should be updated to Tails 3.7, which is scheduled to be released on May 9, 2018. Follow the graphical prompts on your workstations to perform this upgrade.

For the Journalist Workstations and the Admin Workstation, you will also need to update the SecureDrop code (which configures the workstation environment, such as the SecureDrop shortcuts on the Tails desktop). First, ensure that you have set a Tails administrator password. Then, on the workstation to be updated, perform the following steps in a terminal:

```
cd ~/Persistent/securedrop
git fetch --tags
gpg --recv-key "2224 5C81 E3BA EB41 38B3 6061 310F 5612 00F4 AD77"
git tag -v 0.7.0
```

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 you see on the screen of your Admin Workstation. If it does, you can check out the new release:

```
git checkout 0.7.0
```

Finally, run the following commands:

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

Going forward, the SecureDrop Workstation Updater user interface will appear when updates are needed upon boot of a workstation. Users only need to click “Update Now” instead of performing the steps described above. The SecureDrop Workstation Updater looks as follows:
SecureDrop Workstation Updater

Updates Available

SecureDrop workstation updates are available! It is recommended to install them now.

If you don't want to install them now, you can install them the next time you reboot.

You will need to have set a Tails Administration password in the Tails Greeter on boot to complete the update.

When you start your workstation, this window will automatically appear if you have not completed any required updates.
47.2 Configuring SSH Over the Local Network

SecureDrop 0.7.0 ships with the option to SSH into the SecureDrop servers over the local network instead of the Tor network. Please see the detailed documentation for this feature.

47.3 Configuring Email Alerts About New Submissions

SecureDrop 0.7.0 allows you to configure an encrypted daily email alert informing you about whether or not there are new SecureDrop submissions. In order to configure this feature, you will need an email address that should receive alerts, a public GPG key to use for that address, and the fingerprint for that key.

Note that you cannot configure multiple email addresses. If you would like the alerts to be sent to multiple users, you will have to set up a single email alias for this purpose, and all recipients will need to have access to a shared GPG key.

To set up the alerts, on your Admin Workstation, copy the public key (e.g., journalist.pub) to ~/Persistent/securedrop/install_files/ansible-base/.

Within ~/Persistent/securedrop, run ./securedrop-admin sdconfig and follow the prompts. For this functionality, you will be prompted for the following values (default is none, as the functionality is disabled by default):

- Journalist public key (e.g., journalist.pub)
- Journalist GPG key fingerprint
- Journalist email address

Run ./securedrop-admin install and wait for it to complete. The configured email address should now receive an encrypted notification email every 24 hours.

47.4 Configuring Server Reboot Time

SecureDrop 0.7.0 lets you configure the time of day (full hour) at which the SecureDrop servers are rebooted.

To do so, run ./securedrop-admin sdconfig on your Admin Workstation within ~/Persistent/securedrop, and follow the prompts. For this functionality, you will be prompted for “Daily reboot time of the server (24-hour clock)”. Specify the desired local time, e.g., type “18” for 6 PM. Then run ./securedrop-admin install to apply the change to the servers.

47.5 Getting Support

Should you require further support with your SecureDrop installation or upgrade, we are happy to help!

- Community support is available at https://forum.securedrop.org
- The Freedom of the Press Foundation offers training and priority support services. See https://securedrop.org/priority-support/ for more information. If you are already a member of our support portal, please don’t hesitate to open a ticket there.
CHAPTER 48

Upgrade from 0.5.x to 0.6.x

48.1 Updating the Tails Workstations

All Tails drives should be updated to Tails 3.6, released concurrently with SecureDrop 0.6. For the Secure Viewing Station, Admin Workstation and Journalist Workstation, you need to manually upgrade as explained in the Tails documentation.

For the Journalist Workstations and the Admin Workstation, you will also need to update the SecureDrop code using the following manual method:

```
cd ~/Persistent/securedrop
git fetch --tags
git checkout 0.6
gpg --recv-key "2224 5C81 E3BA EB41 3B83 6061 310F 5612 00F4 AD77"
git tag -v 0.6 # Output should include "Good signature"
./securedrop-admin setup
```

From this point forward, you will be able to use the command `securedrop-admin check_for_updates` to check whether updates are available, and `securedrop-admin update` to check for updates and to apply them in a single step.

48.2 Troubleshooting Linux Kernel Issues

The latest Linux kernel (version 4.4.115) will be automatically installed on your SecureDrop servers within 24 hours of the 0.6 release (13th of March at 22:00 UTC). If you are using hardware other than our official hardware recommendations, you face moderate risk of a SecureDrop outage.

If you are experiencing an outage as part of the 0.6 release (e.g. you cannot reach the source interface, you cannot SSH into the servers etc.), you will have to perform the steps outlined below.

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 did not configure Google Authenticator for console access, you will need to use single user mode in order to login to each server.
48.2.1 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 “linux”, replace the word “quiet” with “single”. Note that the word “quiet” may be wrapped, as in the screenshot below:
Press the “F10” key to boot.

### 48.2.2 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, please still 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.

### 48.2.3 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 new kernel from the GRUB boot menu by selecting **Advanced Options for Ubuntu**, pictured below.
Under **Advanced Options**, choose the option with kernel version 3.14.x-grsec. 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 4.x kernel. In that case, we need to temporarily set an older kernel as the default.

### 48.2.4 Roll Back to the Old Kernel

**Important:** The older kernel has reached end of life, so it is of critical importance for the near term 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 3.14.xx-grsec...
```

Take note of its position among the other submenu entries (it will most likely be third). Then edit the GRUB configuration:
sudo vim /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:

```bash
sudo update-grub2
```

Now you can reboot into the old, working kernel.

```bash
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 `3.14.79-grsec`.

Please notify us of the compatibility issue so we can help you resolve it ASAP.

### 48.2.5 Report Compatibility Issues

If you have encountered issues with the kernel upgrade, 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.
48.3 Getting Support

Should you require further support with your SecureDrop installation or upgrade, we are happy to help!

- Community support is available at https://forum.securedrop.org
- Paid support options are provided by Freedom of the Press Foundation. Contact securedrop@freedom.press for more information.
CHAPTER 49

Upgrade from 0.4.x to 0.5.x

**Note:** First follow the instructions in the 0.3 to 0.4 *migration document* if you have not yet updated your *Admin Workstation* to 0.4.

Beginning with SecureDrop 0.5, the source and journalist interfaces are localized. After an unattended upgrade of the application server, these translations are available but are not activated by default.

**Note:** See the *installation documentation* for a list of supported languages.

The steps below should be performed on the *Admin Workstation* associated with your SecureDrop instance.

### 49.1 Pull the Latest Release

Open a **Terminal** and navigate to your SecureDrop directory.

```
$ cd ~/Persistent/securedrop
```

Fetch the latest code, and verify the tag for the latest release (0.5):

```
$ git fetch
$ git tag -v 0.5
```

The output of the above commands should include *Good signature from "SecureDrop Release Signing Key"*. If it does not, please contact us immediately at support@freedom.press.

**Note:** You may also see output from GPG warning you that the key is not certified with a trusted signature. This means that there is not a trust path to the release signing key. As long as you see the fingerprint 2224 5C81 E3BA
Once you’ve verified the latest release, check it out:

```
git checkout 0.5
```

### 49.2 Choose the List of Supported Languages

You need to run the `./securedrop-admin sdconfig` command again, following the same instructions as during the first installation. This will not modify the existing configuration but you will be asked for a list of supported languages because this option did not exist before. You should decide which languages you prefer as explained in the installation documentation.

You should then proceed to update the application server with:

```
./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.

#### 49.2.1 Verify the Source Interface Displays the Selected Languages

The source and journalist interfaces will display the current language with a flag and clicking on the flag will show a menu with all supported languages.
Beginning with SecureDrop 0.4, the use of Tails 3 is required. SecureDrop 0.4 included substantial changes to the Admin tooling used for managing the configuration for the Application and Monitor Servers, and modifies the location of the configuration on Admin Workstation to prevent conflicts in the future.

**Note:** All Admin and Journalist Workstations must be upgraded to Tails 3 for use with SecureDrop 0.4.x. Follow the *Upgrade Tails from 2.x to 3.x* guide for detailed instructions on upgrading if you have not already done so.

The steps below should be performed on **both the Admin and all Journalist Workstations** associated with your SecureDrop instance. You do not need to run these steps on the **Secure Viewing Station**.

### 50.1 Pull the Latest Release

Open a *Terminal* and navigate to your SecureDrop directory.

```
cd ~/Persistent/securedrop
```

Stash your local configuration, fetch the latest code, and verify the tag for the latest release (0.4.1):

```
git stash save "site specific configs"
git fetch
git tag -v 0.4.1
```

The output of the above commands should include **Good signature from "SecureDrop Release Signing Key"**. If it does not, please contact us immediately at support@freedom.press.

**Note:** You may also see output from GPG warning you that the key is not certified with a trusted signature. This means that there is not a trust path to the release signing key. As long as you see the fingerprint **2224 5C81 E3BA**
EB41 38B3 6061 310F 5612 00F4 AD77 displayed and the signature verifies as described above then you can proceed safely.

Once you’ve verified the latest release, check it out, then pop your local configuration back into place:

```
$ git checkout 0.4.1
$ git stash pop
```

## 50.2 Upgrade the Tails Persistence Configuration

SecureDrop 0.4.x provides more convenient tooling for configuring the ATHS info required to access the Journalist Interface. Run the following commands to install the required packages and set up the access to your SecureDrop instance.

```
$ ./securedrop-admin setup
$ ./securedrop-admin tailsconfig
```

### 50.2.1 Clean Up Old Version-Controlled Site Config

The `tailsconfig` task copied the site-specific configuration for your SecureDrop instance to a new location: `install_files/ansible-base/group_vars/all/site-specific`. Beginning with 0.4, manual edits to the inventory are no longer required, as the ATHS information is read automatically from the `app-ssh-aths` and `mon-ssh-aths` files. Therefore you should permanently store any site-specific modifications:

```
$ git stash save "old site-specific configs"
```

During subsequent upgrades to the SecureDrop Admin configuration, you will no longer need to perform `git stash` and `git pop` as described above.

## 50.3 Verify the Upgrades

### 50.3.1 Verify the Journalist Workstation and SVS USB Drives Are Successfully Updated

After you upgrade your Journalist Workstation and Secure Viewing Station, do the following to make sure they were upgraded successfully:

1. Submit a test document to the source interface.
2. Log in to the journalist interface.
3. Download the test document.
4. Transfer the test document over to the SVS.
5. Decrypt the test document.
6. Delete the submission.

If you are able to successfully download and decrypt your test submission, then your upgrade was successful!
50.3.2 Verify the Admin Workstation USB Drive Was Successfully Updated

After you upgrade your Admin Workstation, ensure that you are able to SSH into both servers. Remember you can use the following shortcuts:

```
ssh mon
ssh app
```
Starting with SecureDrop version 0.3.7, SecureDrop’s Tails integration leverages improvements to the Tails OS since the introduction of Tails 2.0. It is critical to upgrade all of your Tails USBs to the latest version of Tails before upgrading SecureDrop to 0.3.7 or later.

**Warning:** Tails 1.x is no longer receiving updates, and is therefore vulnerable to a growing list of security vulnerabilities. We strongly encourage you to upgrade all of your Tails USBs to the latest version of Tails as soon as possible.

Upgrading Tails from 1.x to 2.x must be done manually. Please follow this guide to updating each Tails USB stick used in your SecureDrop instance. Be sure to use the Secure Viewing Station computer so you benefit from its airgap while transferring sensitive data.

**Note:** You will need:

1. A *master Tails USB* running the most recent version of Tails (at least v2.3).
2. A *backup device*, a separate, encrypted USB drive used to store backups of the old Tails sticks.
4. An *airgapped machine* to perform the Tails upgrades. It is ok to reuse the Secure Viewing Station for this task.

An airgapped machine (such as the SVS) is required in order to perform these upgrades 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 all 3 devices at the same time. If you don’t have 3 USB ports available, you can use a USB hub, which may reduce transfer speeds.
51.1 Upgrade Each Tails Device

51.1.1 1. Prepare the Master Tails USB

Because Tails 2.x is incompatible with older versions, you must create a new “master” Tails USB stick for subsequent installations and upgrades to the USB sticks already in-use by your organization. To create this brand-new master Tails, follow the same directions for provisioning the first USB sticks on another networked computer.

Once you’ve created a new Tails 2.x USB, boot into it from your airgapped computer to perform the next steps. At the Tails Greeter screen, be sure to enable admin privileges.

51.1.2 2. Prepare the Backup Device

We will use the Tails Installer to upgrade your Tails 1.x USB to Tails 2.x. While this usually works without any issues, we’re going to start by making backups of the important data on your current Tails USBs, so you can use them for recovery in case something goes wrong.

Tip: While it’s recommended to use a fresh USB stick for any backup operation, to cut down on cost, clutter, and/or waste, you may also repurpose old USB sticks to function as Backup Devices. Note that this process will permanently erase any data stored on the Backup Device.

After logging into the master Tails device, open the Disks Utility by navigating to Applications Utilities Disks. Insert the Backup Device into a USB port. It will appear in the list of disks in the left column. Select it.
Click the button with the interlocking gears icon and choose **Format**...
Fill out the **Format Volume** settings as shown in the screenshot below. There’s no need to overwrite existing data, and doing so can take a long time. You should use a strong passphrase to encrypt the drive.

**Note:** If you plan on using this USB stick as a permanent backup, you will be responsible for retaining this passphrase for the long-haul. If you only want to use this USB as an intermediary backup, and plan on discarding the data after a successful migration, you may discard the passphrase once all steps are completed.
Click **Format**. A dialog box will ask: “Are you sure you want to format the volume?” Click **Format**.

While the drive is being formatted, you will see a spinning progress indicator next to the drive’s name in the left column. Wait until it is done. When it is done, you will see the partition layout has two nested partitions (LUKS and ext4), like this:
You’re ready to start backing up your current Tails USBs.

### 51.1.3 3. Backup a Tails USB

Insert the Tails USB (that you want to back up) into a free USB port.

Mount it by navigating to **Places Computer**, and clicking on the encrypted disk. You will be prompted to enter the passphrase to unlock the disk (the same passphrase you normally use to log into Tails on that USB stick).

Open a Nautilus window with admin privileges by navigating to **Applications System Tools Root Terminal**. At the terminal prompt, simply type `nautilus`.
The Nautilus window should show both the Backup Device and the TailsData partition as mounted.

Copy the all data from the TailsData partition onto the Backup Device except:

- **persistence.conf** In older versions of Tails, this file might have slightly different directives in it that could temporarily brick a Tails 2.x USB.

- **claws-mail** Claws Mail is no longer included in Tails. The OS uses Icebird instead. Some users might not have this folder, so if you don’t see it there, do not be alarmed.
Ensure that all critical data has been successfully copied. Specifically, be sure the gnupg, bookmarks, and Persistent folders are completely copied. Any loss of data from these folders could prevent users from accessing submissions.

**Tip:** Create subdirectories for each USB drive (Admin, Journalist, and SVS) within the Backup Device. Not only will doing so speed up the upgrade process, it will also provide you with long-term encrypted backups of the USB devices. In the event of a lost or stolen drive, you can restore access via this encrypted backup device.

Once data are correctly copied, unmount the TailsData partition.
51.1.4 4. Upgrade a Tails USB

With the Admin/Journo/SVS Tails USB still inserted in the machine, navigate to Applications Tails Tails Installer and select the Upgrade by cloning option.

Select the Tails 1.x USB that you wish to upgrade from the drop-down menu labeled Target Device. If it is the only other USB plugged in to the computer, it should be automatically selected.
The clone process will take a few minutes, and will display a message once it is complete. If you see an error message about the device not being ready, try unplugging and remounting the Tails device you’re trying to upgrade.

51.1.5 5. Re-Install the Automatic Tails Configuration

Note: This step is only applicable to the Admin Workstation and Journalist Workstation Tails USBs. If you are upgrading the Secure Viewing Station Tails USB, you can skip this step.

Shut down the Tails USB on the airgapped computer and move it to the computer you normally use it on. Boot into each newly upgraded Tails USB, enabling persistence, and setting a root password. Confirm that the persistent files are present on the upgraded Tails USB. If they are not, or something else went wrong, see Troubleshooting.
Now that you have successfully upgraded to Tails 2.x with your persistence intact, the final step is to re-install the Tails automatic configuration from the latest version of SecureDrop (0.3.7 or later). The Tails auto-configuration was originally set up during installation in Configure the Admin Workstation Post-Install and Onboard Journalists. There were enough changes in Tails 2.x that we had to update various aspects of the auto-configuration to work properly on it, which is why you need to re-install.

Once you’ve ensured that you’re running SecureDrop 0.3.7 or later, you can re-install the Tails auto-configuration:

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

This is the same process as described in Configure the Admin Workstation Post-Install (for the Admin Workstation) and Onboard Journalists (for the Journalist Workstations). If you have questions, consult that documentation first.

When you’re done, repeat this final step on the rest of the Tails devices. Once you have re-installed the Tails auto-configuration on all of the Tails devices, move on to the **Finishing up** section below.

### 51.2 Finishing Up

#### 51.2.1 Verify All Devices Are Working

Shut down each Tails USB on the airgapped computer and move it to the computer you normally use it on. Boot into each newly upgraded Tails USB, enabling persistence. Confirm that the persistent files are present and that your workflow is unaffected.

As a test, consider submitting a test submission, downloading it on the Journalist Workstation, and finally decrypting it on the SVS. If you are able to decrypt the submission successfully, you have verified that the Journalist Workstation and SVS are working correctly after the upgrade.

To test the Admin Workstation, make sure you can still SSH into the servers:

```
$ ssh <username>@<*Application Server* .onion address> hostname
app
$ ssh <username>@<*Monitor Server* .onion address> hostname
mon
```

**Tip:** If you forgot, your SSH username is in `install_files/ansible-base/group_vars/all/site-specific` as the value of the `ssh_users` variable. The .onion addresses for SSH for each server are in `install_files/ansible-base/app-ssh-aths` and `install_files/ansible-base/mon-ssh-aths`, respectively.

**Tip:** Consider retaining the encrypted backup drive as a disaster recovery device. Document the passphrase in the Admin Workstation KeePassX database, and store the physical Backup Device in a locked safe or other secure location.

#### 51.2.2 Wipe the Backup Device

If you do not have a secure location for storing the backups, or already have other backups, you should wipe the Backup Device. There is a lot of debate over the best way to do this, but we think it’s sufficient to simply overwrite it with random data a couple of times. Since the Backup Device is encrypted with LUKS, which employs a number of anti-forensic-recovery techniques, this should be enough to prevent forensic recovery.
First, find the path to the Backup Device. You can find the path with the Disks application, selecting the drive in the left column, and looking at the Device entry. It is usually a string that starts with /dev/sd.

**Warning:** Make sure you use the correct path for the Backup Device in the next command! Otherwise, you run the risk of irreversibly wiping a different drive on the system, such as the Tails USB you are running.

To overwrite the Backup Device, open a Terminal and run:

```
dd if=/dev/urandom of=<path to Backup Device>
```

Re-run this command at least twice. Each run will take a while.

If you want to reuse the drive for another purpose, use the Disks utility to reformat it appropriately.

**Note:** While it probably isn’t necessary to physically destroy a Backup Device (because it’s encrypted, and LUKS is designed to thwart forensic recovery), if you’re really paranoid you can additionally smash the device with a hammer until the chips containing its flash memory are broken up, then dispose of the pieces in the garbage.

### 51.3 Troubleshooting

The steps described above should cleanly update your Tails devices without issue. In the event that one or more of your upgraded Tails USBs are not working as expected, don’t worry: you can still manually restore from the Backup Device you created. (Isn’t it great to have backups?)

#### 51.3.1 1. Restore Data from the Backup Device

On the same airgapped machine, boot up the Tails USB stick you want to restore, with both persistence and admin privileges. Insert your Backup Device into a free USB port, and mount it by navigating to Places Computer, and clicking on the encrypted disk. You will be prompted to enter its passphrase.

Open a Nautilus window with admin privileges by navigating to Applications System Tools Root Terminal. At the terminal prompt, simply type `nautilus` and hit Enter. Type `ctrl + l`, type `/live/persistence/TailsData_unlocked`, and hit Enter to navigate there.
Open a new tab in Nautilus (ctrl + t) and navigate to your Backup Device. Drag and drop the backup data from your Backup Device onto the TailsData_unlocked tab.

When copying a folder, select the Apply this action to all files option and click Merge to apply to all subfolders. Then you might have to select again the Apply this action to all files option and click Replace to apply to all files.

In a root terminal, or as sudo, execute the following command to fix the ownership of your personal files:

\section*{51.3. Troubleshooting}
find /live/persistence/TailsData_unlocked/ -uid 1000 -exec chown -R 1000:1000 '{}' \\;

51.3.2 2. Verify the Restored Data

Shut down, and reboot the Tails USB. Now that you’ve restored the files, you should re-do the post-upgrade verification to make sure everything is working correctly.
CHAPTER 52

Upgrade Tails from 2.x to 3.x

52.1 Why You Should Upgrade

Starting with SecureDrop version 0.4, we require users update to Tails version 3.0 or later. Upgrading Tails from 2.x to 3.x must be done manually. This guide will show you how to upgrade each Tails USB stick used in your SecureDrop instance.

52.2 What You Need


2. An airgapped machine to perform the Tails upgrades. The Secure Viewing Station may be used for this task.

3. Two USB drives: one to create a new master Tails 3.x USB and one to backup the data on your current SecureDrop Tails USB sticks.

**Warning:** An airgapped machine (such as the Secure Viewing Station) is required in order to perform these upgrades 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 Tails drive you wish to upgrade, the *master Tails USB* drive, 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.
52.3 1. Prepare the Master Tails USB

Because Tails 3.x is incompatible with older versions, you must create a new “master” Tails USB stick for subsequent installations and upgrades to the USB sticks already in use by your organization.

To create this brand new master Tails, follow the same directions for provisioning the first USB sticks on another networked computer.

Once you’ve created a new Tails 3.x USB, boot into it from your airgapped computer to perform the remaining steps.

At the Tails Greeter screen, enable admin privileges by setting a root password. In Tails 3.x, you do this by clicking the + button, then navigating to Additional Settings Administration Password.

52.4 2. Backup the Tails Drives

Note: The steps in this section should be performed for each Secure Viewing Station, Journalist Workstation, and Admin Workstation USB drive in your organization.

Before you upgrade your Tails drives, you should backup the data in case something goes wrong.

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**: Make sure you use a strong passphrase if this is a long term backup drive.

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 usually use when you enable Persistence on that device:
You should now have both the Backup and TailsData partition to be backed up mounted and ready to access.
Open a Nautilus window with admin privileges by going to **Applications System Tools Terminal**.
Type `gksu nautilus` at the terminal prompt and hit enter. You’ll need to type your admin password.
Note: When you run `gksu nautilus`, you may run into an error where Nautilus complains that it can’t create a required folder. If that happens, just click OK and continue normally.

If a Nautilus window doesn’t come up, it might be because an admin password wasn’t set. If that’s the case, you’ll need to restart and set an admin password before continuing.

Warning: Make sure you use keep the Terminal window open while you perform the backups. Otherwise, the Nautilus window will close.

Make sure you create a directory on the backup drive to store the data from the drive you are backing up:
Copy over everything in the TailsData partition to the relevant folder on the Backup drive. You can simply drag to select all the files and then copy and paste them to the relevant folder on the Backup drive.

In particular, ensure gnupg and Persistent have been successfully copied over. These files are critical for decrypting submissions.

Once complete, unmount the TailsData partition.

Repeat these steps for every device, making a new folder on the backup device for each device you backup.

Finally, once you have completed the steps described in this section for each USB drive, unmount the Backup partition and store the drive somewhere safely.

### 52.5 3. Upgrade the Tails Drives

**Note:** The steps in this section should be performed for each Secure Viewing Station, Journalist Workstation, and Admin Workstation USB drive in your organization.

Next you will upgrade each drive.

Begin by inserting the drive you wish to upgrade into the machine.

Navigate to Applications Tails Tails Installer.
Click Upgrade by cloning.
Make sure the correct drive is selected.
Click **Install Tails**.

A dialog box will appear asking you to **Please confirm your device selection**.
Click **Yes** to proceed with the installation.

**Note:** The upgrade can take quite a bit of time, so please be patient!

Once complete, you should see a success message:
52.6 4. Upgrade KeePassX Database

Your password databases will be in KeePass 1 database format (a file that ends in .kdb). You should upgrade them to the new format by following these steps:

1. Open KeePassX.
2. Navigate to Database and then Import KeePass 1 database.
3. Select your password database and click Open.
4. Put in a master password if necessary to open the database.

5. Then navigate to **Database** and then **Save database as** to save the database in its new format (a file ending in `.kdbx`) in the same folder as the previous database.

### 52.7 5. Upgrade the Secure Viewing Station

Due to a change in Tails 3, if you wish to preserve the names of files when decrypting, you’ll need to apply the following fix by opening a **Terminal** on the **Secure Viewing Station** and typing the following commands:

```
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.

### 52.8 6. Upgrade SecureDrop to 0.4.x

Now that you’ve upgraded the Tails workstation to Tails 3, follow the **0.4.x Upgrade Guide** to configure the Tails environment to access your SecureDrop instance. You will need to perform further upgrade steps for the **Admin** and **Journalist Workstations**.

After upgrading to 0.4.x, you should move your backup drive to a safe location (if you used a strong passphrase). Else, you should destroy the backup drive following the instructions here.

### 52.9 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**.
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.

53.1 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:

53.1.1 Bugs

• High-priority bugs
• Middle-priority bugs
• Low-priority bugs

53.1.2 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

53.1.3 Development Environment

• 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.

53.2 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.
53.2.1 Documentation Issues

- High-priority
- Middle-priority
- Low-priority

53.3 DevOps

The SecureDrop web site and the GitHub repository are controlled and maintained by Freedom of the Press Foundation employees.

53.4 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 grsecurity kernels and pending bugs and tasks

53.5 Translators

All are kindly invited to help translate SecureDrop using the Weblate interface. We provide a detailed guide to use as reference for details such as the meaning of placeholders, etc. Feel free to reach out on the translation section of the forum for help. Non-English forum discussions are welcome to help facilitate translations.

53.6 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

53.7 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.
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.

54.1 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.

54.2 Quick Start

The Docker based environment is suitable for developing the web application and updating the documentation.

54.2.1 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.

54.2.2 macOS

Install Docker.

54.2.3 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.

54.2.4 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
```

54.2.5 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
```
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

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('JHCOGO7VCER3J4L').now()
'422038'
```

## 54.3 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.

### 54.3.1 Ubuntu or Debian GNU/Linux

**Note:** Tested on: Ubuntu 16.04 and Debian GNU/Linux stretch

```bash
sudo apt-get update
sudo apt-get install -y build-essential libssl-dev libffi-dev python-dev \
    dpkg-dev git linux-headers-$\(\text{uname} -r\) 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](https://www.vagrantup.com/downloads/) 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](https://github.com/secureDROP-dev/SecureDROP/issues/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`:
SecureDrop Documentation, Release 0.12.0~rc1

```bash
sudo apt-get install virtualenvwrapper
source /usr/share/virtualenvwrapper/virtualenvwrapper.sh
mkvirtualenv -p /usr/bin/python2 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.

### 54.3.2 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.

```bash
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`.

### 54.3.3 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 your local machine:

```bash
git clone git@github.com:<your_github_username>/securedrop.git
```

### 54.3.4 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/develop-requirements.txt

**Note:** You will need to run this everytime new packages are added.
Making a PR to SecureDrop

55.1 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`

55.2 Make Your Changes and Push to the Fork

55.2.1 Create a Branch

Create a branch on which you make your changes.

   `git checkout -B change-one`

55.2.2 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>"
```

### 55.2.3 Pull the Upstream Changes

We get any updates made in the upstream repository.
```
git pull upstream develop
```

### 55.2.4 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.

### 55.2.5 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>
```

### 55.3 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.
Development of Securedrop-Admin in the Admin Directory

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:

```
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:

```
bin/dev-shell tox -e flake8
```

Run only one test foobar with:

```
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:

```
$ make
Makefile for developing and testing securedrop-admin.
Subcommands:

help Print this message and exit.
test  Run tox
update-pip-requirements Updates all Python requirements files via pip-compile.
```
Development of SecureDropUpdater in the journalist_gui Directory

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.

### 57.1 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.

### 57.2 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.
SecureDrop Documentation, Release 0.12.0-rc1

$ sudo apt install qtcreator python3-pyqt5

If we make any changes to the UI, we will have to use pyuic5 command to update the corresponding Python code.

$ pyuic5 journalist_gui/mainwindow.ui -o journalist_gui/updaterUI.py

### 57.3 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>` element.

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:

$ 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.

### 57.4 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:

$ 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.

58.1 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!

58.2 How to Find Help

If you would like to report a problem [submit a new issue](https://github.com/freedomofpress/securedrop-client/issues).

If you’d like to chat with other developers working on the client drop into our [Gitter chat channel for the project](https://gitter.im/freedomofpress/securedrop-client).
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.

## 58.3 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.
- `models.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.

## 58.4 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.

58.5 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.

### 59.1 Versioning

The API is versioned and we are currently using version 1. This is set via the base URL, which is:

```
/api/v1/
```

### 59.2 Content Type

Clients shall send the following headers:

```
'Accept': 'application/json',
'Content-Type': 'application/json'
```

### 59.3 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
eyJhbGciOiJIUzI1NiIsImV4cCI6MTUzMDU4NjU4MiwiaWF0IjoxNTMwNTc5MzgyfQ.eyJpZCI6MX0.PfcLmK1Dq5VCIANo-1Jbmu02yCl2VcT8qf9fIZaTcm
```

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, but note that clients should use the expiration time provided in the response to determine when the token will expire. After the token expires point, users must login again. Clients implementing logout functionality should delete tokens locally upon logout.

### 59.4 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."}
```

### 59.5 Endpoints

#### 59.5.1 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",
 replies_url": "/api/v1/replies",
 token_url": "/api/v1/token"
}```
59.5.2 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):

```
{
  "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": {
        "public": "-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBFtEA0YBEAChcaDWfnLvMNDypxF+YhNI7/

Oz3NU4G5+MYHstD3m4Cdcwdvo+S6E66B4h/

9xWWtJLzBMmRNBrCpfnym8id1QyNadNzPPYyk2Xt6Xs9R2aH03sdB8nXVx07FwYmMzNa3U1Rg6kb0EUwzNDOW0jeramutp1c0\n
449DoJFeRWar1GnhssfALVjhizf1sL1NXCybjaCqkN

qVMsKmYpDZwIXOFF77JWNJ2n/

N3EpGhqlppjjaJ4LNPnXstCmqAk50ckP5P1hUH2q}\n
6YWtPvyQgO9iV5cWlgL2KQd53V02+6+raANAEe23yAnJWc7HLCkeB4290\n
htq\n
BFU2xRrV22UxXjhdGVKe1te1Sa8TTD0EOc3RJNQU9KREhP9\n
BBMCgApBQJbrRANGahs\n
6BwaJCaCdaEGFOgQCQ0LBBYCAwEC1bHc4ACkqkDDX++nnxkd1+CA/9\n
G0g3Xm3e2pyW+iuxKC/g0jIk/K

pxk\n
nZpN5F5d1bLTeBKMMmMy2d4ycC/

7br\n
r3Q0D\n
\n
9\n
40\n
l0\n
14\n
3\n
6\n
n\n
x\n
\n
-----END PGP PUBLIC KEY BLOCK-----

----E-----BEGIN PGP PUBLIC KEY BLOCK----

mQINBFtEA0YBEAChcaDWfnLvMNDypxF+YhNI7/

Oz3NU4G5+MYHstD3m4Cdcwdvo+S6E66B4h/

9xWWtJLzBMmRNBrCpfnym8id1QyNadNzPPYyk2Xt6Xs9R2aH03sdB8nXVx07FwYmMzNa3U1Rg6kb0EUwzNDOW0jeramutp1c0\n
449DoJFeRWar1GnhssfALVjhizf1sL1NXCybjaCqkN

qVMsKmYpDZwIXOFF77JWNJ2n/

N3EpGhqlppjjaJ4LNPnXstCmqAk50ckP5P1hUH2q}\n
6YWtPvyQgO9iV5cWlgL2KQd53V02+6+raANAEe23yAnJWc7HLCkeB4290\n
htq\n
BFU2xRrV22UxXjhdGVKe1te1Sa8TTD0EOc3RJNQU9KREhP9\n
BBMCgApBQJbrRANGahs\n
6BwaJCaCdaEGFOgQCQ0LBBYCAwEC1bHc4ACkqkDDX++nnxkd1+CA/9\n
G0g3Xm3e2pyW+iuxKC/g0jIk/K

pxk\n
nZpN5F5d1bLTeBKMMmMy2d4ycC/

7br\n
r3Q0D\n
\n
9\n
40\n
l0\n
14\n
3\n
6\n
n\n
x\n
\n
-----END PGP PUBLIC KEY BLOCK-----

----E-----BEGIN PGP PUBLIC KEY BLOCK----

mQINBFtEA0YBEAChcaDWfnLvMNDypxF+YhNI7/

Oz3NU4G5+MYHstD3m4Cdcwdvo+S6E66B4h/

9xWWtJLzBMmRNBrCpfnym8id1QyNadNzPPYyk2Xt6Xs9R2aH03sdB8nXVx07FwYmMzNa3U1Rg6kb0EUwzNDOW0jeramutp1c0\n
449DoJFeRWar1GnhssfALVjhizf1sL1NXCybjaCqkN

qVMsKmYpDZwIXOFF77JWNJ2n/

N3EpGhqlppjjaJ4LNPnXstCmqAk50ckP5P1hUH2q}\n
6YWtPvyQgO9iV5cWlgL2KQd53V02+6+raANAEe23yAnJWc7HLCkeB4290\n
htq\n
BFU2xRrV22UxXjhdGVKe1te1Sa8TTD0EOc3RJNQU9KREhP9\n
BBMCgApBQJbrRANGahs\n
6BwaJCaCdaEGFOgQCQ0LBBYCAwEC1bHc4ACkqkDDX++nnxkd1+CA/9\n
G0g3Xm3e2pyW+iuxKC/g0jIk/K

pxk\n
nZpN5F5d1bLTeBKMMmMy2d4ycC/

7br\n
r3Q0D\n
\n
9\n
40\n
l0\n
14\n
3\n
6\n
n\n
x\n
\n
-----END PGP PUBLIC KEY BLOCK-----

----E-----BEGIN PGP PUBLIC KEY BLOCK----

mQINBFtEA0YBEAChcaDWfnLvMNDypxF+YhNI7/

Oz3NU4G5+MYHstD3m4Cdcwdvo+S6E66B4h/

9xWWtJLzBMmRNBrCpfnym8id1QyNadNzPPYyk2Xt6Xs9R2aH03sdB8nXVx07FwYmMzNa3U1Rg6kb0EUwzNDOW0jeramutp1c0\n
449DoJFeRWar1GnhssfALVjhizf1sL1NXCybjaCqkN

qVMsKmYpDZwIXOFF77JWNJ2n/

N3EpGhqlppjjaJ4LNPnXstCmqAk50ckP5P1hUH2q}\n
6YWtPvyQgO9iV5cWlgL2KQd53V02+6+raANAEe23yAnJWc7HLCkeB4290\n
htq\n
BFU2xRrV22UxXjhdGVKe1te1Sa8TTD0EOc3RJNQU9KREhP9\n
BBMCgApBQJbrRANGahs\n
6BwaJCaCdaEGFOgQCQ0LBBYCAwEC1bHc4ACkqkDDX++nnxkd1+CA/9\n
G0g3Xm3e2pyW+iuxKC/g0jIk/K

pxk\n
nZpN5F5d1bLTeBKMMmMy2d4ycC/

7br\n
r3Q0D\n
\n
9\n
40\n
l0\n
14\n
3\n
6\n
n\n
x\n
\n
-----END PGP PUBLIC KEY BLOCK-----

"sources"

"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"

(continues on next page)

59.5. Endpoints 347
59.5.3 Individual Source [/sources/<source_uuid>] 

Requires authentication

An object representing a single source.

Response 200 (application/json):

```json
{
    "add_star_url": "/api/v1/sources/f086bd03-1c89-49fb-82d5-00084c17b4ce/add_star",
    "interaction_count": 2,
    "is_flagged": false,
    "is_starred": false,
    "journalist_designation": "navigational firearm",
    "key": {
        "public": "-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBFtEA0sBEACsJK+UPZoemYts+L+4JnhaRXjqixMO2BDJEueiGq2aQ0CEI4pzn\nmNq5Xn/ZjHChnh/3AEc/Svv1IpA8RH4pBiF8ri3C1IdKFmjwo6a69YNjOj12\n7p+BIHLCrehIY0Z8Q7yYxOuNR9pVry7ioNNeKJaoBcXIffsL5n5QIfVS9rX+\nnNSN5UF+yEB/90FFywwHaHlVhYLMmnlkiU7dAbxowJxbw7ShN/Sp1LmIV/4UYHIFt\n\n90uS5Fm9uoGyRQoj OwtwC7bwARAQAB\n
mOenYe6dCXgJN128Bb7v3MaZ9Wu+5qy+3RCDeGvThB1\nmwZ8tja6o6z9r1q9/

Lexe07xjJLE4GRG0i2v8WLFPv+v1ULpsGCnscn8ipEnwrtY9Mw3\nbioArNhrp0kCvn6aiU18Lb3n157Fp9dKfBmP7evf0DcEvF9

YpccCN2JP18roovw1Brr+Lhyjky5UEH1XZkEh6mO60q1z\n\nG8Jb2BccC190nhL3ijsGSVPqIuEN0VuNa1Tfzf1bhJ/\nAYMbCqGRSU3a0KNWRa+\n\nNE25E5VqKwtal2Bvd2tkJbS335nCbkuIjVgMl9y9j2dT58ZegmPwCzA670nDovCy

b0uvYbhD04N/

ZpHjG2ja\n2u8uQ89v9BjGPr5u2W40f842u17J6F65x7+phnoy6ayXCV0fwgJ3g85dpUUPITA\n
----END PGP PUBLIC KEY BLOCK-----\n
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBFtEA0sBEACsJK+UPZoemYts+L+4JnhaRXjqixMO2BDJEueiGq2aQ0CEI4pzn\nmNq5Xn/ZjHChnh/3AEc/Svv1IpA8RH4pBiF8ri3C1IdKFmjwo6a69YNjOj12\n7p+BIHLCrehIY0Z8Q7yYxOuNR9pVry7ioNNeKJaoBcXIffsL5n5QIfVS9rX+\nnNSN5UF+yEB/90FFywwHaHlVhYLMmnlkiU7dAbxowJxbw7ShN/Sp1LmIV/4UYHIFt\n
90uS5Fm9uoGyRQoj OwtwC7bwARAQAB\n
mOenYe6dCXgJN128Bb7v3MaZ9Wu+5qy+3RCDeGvThB1\nmwZ8tja6o6z9r1q9/

Lexe07xjJLE4GRG0i2v8WLFPv+v1ULpsGCnscn8ipEnwrtY9Mw3\nbioArNhrp0kCvn6aiU18Lb3n157Fp9dKfBmP7evf0DcEvF9

YpccCN2JP18roovw1Brr+Lhyjky5UEH1XZkEh6mO60q1z\n\nG8Jb2BccC190nhL3ijsGSVPqIuEN0VuNa1Tfzf1bhJ/\nAYMbCqGRSU3a0KNWRa+\n\nNE25E5VqKwtal2Bvd2tkJbS335nCbkuIjVgMl9y9j2dT58ZegmPwCzA670nDovCy

b0uvYbhD04N/

ZpHjG2ja\n2u8uQ89v9BjGPr5u2W40f842u17J6F65x7+phnoy6ayXCV0fwgJ3g85dpUUPITA\n
----END PGP PUBLIC KEY BLOCK-----\n

"type": "PGP"
},
"last_updated": "2018-07-10T00:52:25.696391Z",
"number_of_documents": 0,
"number_of_messages": 2,
"remove_star_url": "/api/v1/sources/f086bd03-1c89-49fb-82d5-00084c17b4ce/remove_star",
"replies_url": "/api/v1/sources/f086bd03-1c89-49fb-82d5-00084c17b4ce/replies",
"submissions_url": "/api/v1/sources/f086bd03-1c89-49fb-82d5-00084c17b4ce/submissions",
"url": "/api/v1/sources/f086bd03-1c89-49fb-82d5-00084c17b4ce",
"uuid": "f086bd03-1c89-49fb-82d5-00084c17b4ce"
}
}
```
["is_starred": false,
"journalist_designation": "validated benefactress",
"key": {
  "type": "PGP",
  "public": "-----BEGIN PGP PUBLIC KEY BLOCK-----
mQINBFtEA0YBEAChcaDWfnLvMNDypxF+YhNI/
P0wYw7+kGYATcRA8iVQNHQcHdMVyLd9/s5M9bO75
9xWWTJLxBmMNBrCpfny8id1QyNad\n2PPYk20t6Xs9RZaH03sd8nXVx07FwYmMzNa3U1Rg6kb0EUwzND0W0jaramutp1c0
449DoYFeWR1GnhssfAIvijhizf1sf1NXCybACk7\nqVMsKnDzWzIXOPF7jNN7j
N3EpGhq1ppjAJ4LNPNxsTgCmkA5okcPSPlHjU2g\n6NWTVpYrGQ9iV5cWgL2KdQ35v02+6+raANAEe23yAnW9c7HLRckeB4290
Htq\nNRgUelhDdbhr8kzdmdf9WVw/
Tf373FE5FL2mQ7EVIxq44H4vwhWRFjpQKWrRzbsq\n
[continued on next page]
"submission_url": "/api/v1/sources/598b859c-72c7-4e53-a68c-b725eb514241/submission/4c2e701c-70d2-4cb5-87c0-de59c2ebbc62",
    "uuid": "4c2e701c-70d2-4cb5-87c0-de59c2ebbc62"
  },
  {
    "download_url": "/api/v1/sources/598b859c-72c7-4e53-a68c-b725eb514241/submission/c2e00865-8f75-444a-b5b4-88424024ce69/download",
    "filename": "2-dejected_respondent-msg.gpg",
    "is_read": false,
    "size": 604,
    "source_url": "/api/v1/sources/598b859c-72c7-4e53-a68c-b725eb514241",
    "submission_url": "/api/v1/sources/598b859c-72c7-4e53-a68c-b725eb514241/submission/c2e00865-8f75-444a-b5b4-88424024ce69",
    "uuid": "c2e00865-8f75-444a-b5b4-88424024ce69"
  }
]
}

Get a single submission associated with a source [GET]

Requires authentication.

GET /api/v1/sources/<source_uuid>/submissions/<submission_uuid>

Response 200 (application/json):

{
    "download_url": "/api/v1/sources/598b859c-72c7-4e53-a68c-b725eb514241/submission/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/submission/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):

[  
    "filename": "3-famished_sheep-reply.gpg",
    "is_deleted_by_source": false,
    "journalist_username": "journalist",
    "journalist_uuid": "a2405127-1c9e-4a3a-80ea-95f6a71e5738",
   
]
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_uuid": "a2405127-1c9e-4a3a-80ea-95f6a71e5738",
    "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.

```plaintext
DELETE /api/v1/sources/<source_uuid>/replies/<reply_uuid>
```

Response 200:

```json
{
 "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.

```plaintext
POST /api/v1/sources/<source_uuid>/replies
```

with the reply in the request body:

```json
{
 "uuid": "0bc588dd-f613-4999-b21e-1cebbd9adc2c",
 "reply": "-----BEGIN PGP MESSAGE-----[...]
```

Response 201 created (application/json):

```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.

```plaintext
DELETE /api/v1/sources/<source_uuid>/submissions/<submission_uuid>
```

Response 200:
Download a submission [GET]

Requires authentication.

```plaintext
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:c757c5aa263dc4a5a2bca8e7fe973367dbd2clae6c780d19c0ba499e6b1b81efa"
```

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.

```plaintext
DELETE /api/v1/sources/<source_uuid>
```

Response 200:

```json
{
  "message": "Source and submissions deleted"
}
```

Star a source [POST]

Requires authentication.

```plaintext
POST /api/v1/sources/<source_uuid>/star
```

Response 201 created:

```json
{
  "message": "Star added"
}
```

Remove a source [DELETE]

Requires authentication.

```plaintext
DELETE /api/v1/sources/<source_uuid>/star
```

Response 200:
Flag a source [POST]

Requires authentication.

POST /api/v1/sources/<source_uuid>/flag

Response 200:

```
{
  "message": "Source flagged for reply"
}
```

59.5.4 Submission [/submissions]

Get all submissions [GET]

Requires authentication. This gets details of all submissions across sources.

GET /api/v1/submissions

Response 200:

```
{
  "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": "/api/v1/sources/1ed4c191-c6b1-463b-92a5-102deaf7d40a/submissions/a93d4123-a984-4740-9849-772c30694bab/download",
      "filename": "2-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/a93d4123-a984-4740-9849-772c30694bab",
      "uuid": "a93d4123-a984-4740-9849-772c30694bab"
    },
    {
      "download_url": "/api/v1/sources/598b859c-72c7-4e53-a68c-b725eb514241/submissions/4ce2701c-70d2-4cb5-87c0-de59c2ebbc62/download",
      "filename": "1-dejected_respondent-msg.gpg",
      "is_read": false,
      "size": 604,
      "source_url": "/api/v1/sources/598b859c-72c7-4e53-a68c-b725eb514241",
      "submission_url": "/api/v1/sources/598b859c-72c7-4e53-a68c-b725eb514241/submissions/4ce2701c-70d2-4cb5-87c0-de59c2ebbc62",
      "uuid": "4ce2701c-70d2-4cb5-87c0-de59c2ebbc62"
    }
  ]
}
```
59.5.5 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",
            "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"
        },
        {
            "filename": "4-famished_sheep-reply.gpg",
            "is_deleted_by_source": false,
            "journalist_username": "journalist",
            "journalist_uuid": "a2405127-1c9e-4a3a-80ea-95f6a7e5738",
            "reply_url": "/api/v1/sources/f381dbb4-4bb5-451a-801a-e961461af6e5/replies/2863e3ec-66c8-4b74-ba43-615c805be4da",
            "size": 1116,
            "source_url": "/api/v1/sources/f381dbb4-4bb5-451a-801a-e961461af6e5",
            "uuid": "2863e3ec-66c8-4b74-ba43-615c805be4da"
        }
    ]
}
```
356 Chapter 59. Journalist Interface API

59.5.6 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-95f6a7e5738"
}
```
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.

### 60.1 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](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 staging:

```
make build-debs
vagrant up /staging/
vagrant ssh app-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
<?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 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-source-ths`
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 via Vagrant for staging hosts, so you can use `vagrant ssh app-staging` and `vagrant ssh mon-staging` to start an interactive session on either server.

### 60.2 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.

#### 60.2.1 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:

```bash
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:

```bash
sudo addgroup libvirtd
sudo usermod -a -g libvirtd $USER
```

Install the required Vagrant plugins for converting and using libvirt boxes:

```bash
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:

```bash
libvirtd --version
kvm-ok
```
Debian 9 setup

Install Vagrant, libvirt, QEMU, and their dependencies:

```bash
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:

```bash
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:

```bash
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:

```bash
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 bento/14.04 Vagrant box from virtualbox to libvirt format:

```bash
vagrant box add --provider virtualbox bento/ubuntu-14.04
vagrant mutate bento/ubuntu-14.04 libvirt
```

You can now use the libvirt-backed VM images to develop against the SecureDrop multi-machine environment.

**60.2.2 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:
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
```

### 60.2.3 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:

```
ssh_users: "vagrant"
monitor_ip: "10.0.1.5"
monitor_hostname: "mon-prod"
app_hostname: "app-prod"
app_ip: "10.0.1.4"
```

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:

```
vagrant up /prod/
vagrant ssh app-prod
sudo su
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-source-ths
- app-journalist-aths
- app-ssh-aths
- mon-ssh-aths
60.2.4 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.
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.

### 61.1 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.

#### 61.1.1 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.
SecureDrop Documentation, Release 0.12.0~rc1

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.

61.1.2 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.

61.1.3 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:

```python
--- /usr/lib/python2.7/dist-packages/tails_installer/creator.py 2018-01-22 14:59:40 0000000000 +0100
+++ /usr/lib/python2.7/dist-packages/tails_installer/creator.py.mod 2018-03-05 05:15:00 0000000000 -0800
@@ -595,16 +595,6 @@
class LinuxTailsInstallerCreator(TailsInstallerCreator):
     % 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",
-            % {'device': data['udi'], 'interface': iface}
-        )
-    continue
-
-    # Skip optical drives
```

(continues on next page)
if data['is_optical'] and self.opts.force != data['device']:
    self.log.debug('Skipping optical device: %s' % data['device'])

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.

### 61.1.4 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.

### 61.1.5 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:

```bash
--- /usr/share/perl5/Tails/Persistence/Setup.pm 2017-06-30 09:56:25.000000000 +0000
+++ /usr/share/perl5/Tails/Persistence/Setup.pm.mod 2017-07-20 07:17:48.472000000 +0000
@@ -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',
+needs_drive_arg => 1,
   message => $self->encoding->decode(gettext("Device %s is optical.")),
   must_be_false => 1,
   },
   {
       method => 'started_from_device_installed_with_tails_installer',
+needs_drive_arg => 1,
   message => $self->encoding->decode(gettext("Device %s was not created using Tails Installer.")),
   }
   
   my @checks = (
```

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.

### 61.1.6 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:

```
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.

### 61.1.7 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.

### 61.1.8 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.

### 61.1.9 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.

### 61.2 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.
61.2.1 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.

61.2.2 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.
CHAPTER 62

Upgrade Testing using Molecule

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 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.

62.1 Upgrade testing using locally-built packages

First, build the app code packages and create the environment:

```
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.

### 62.2 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.

### 62.3 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. `git checkout <version>`
2. `make vagrant-package`
3. `mv molecule/vagrant_packager/build/app-staging{,_<version>}.box`
4. mv molecule/vagrant_packager/build/mon-staging{,_<version>}.box

5. sha256sum molecule/vagrant_packager/build/*.box

6. Manually update molecule/vagrant_packager/box_files/*.json with new version information, including URL and checksum.

7. cd molecule/vagrant_packager && ./push.yml to upload to S3

8. 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.
it will build!!!

### 63.1 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.

### 63.2 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.

### 63.3 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.

### 63.4 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:

```bash
ln -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:

```bash
pip install -r securedrop/requirements/develop-requirements.txt
```

#### 63.4.1 Python

All Python code should be flake8 compliant. You can run `flake8` locally via:

```bash
make flake8
```

#### 63.4.2 Shell

All Shell code (e.g. `bash`, `sh`) should be shellcheck compliant. You can run `shellcheck` locally via:

```bash
make shellcheck
```

For reference, consult the shellcheck wiki for detailed explanations of any reported violations.

#### 63.4.3 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:

```bash
make html-lint
```

#### 63.4.4 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:

```bash
make yamllint
```
63.5 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.

63.5.1 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.
63.5.2 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.

```bash
$ python3 -m venv ../.py3
$ source ../.py3/bin/activate
$ pip install mypy
$ mypy --py2 securedrop
```

63.6 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.

63.7 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.

63.8 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.
64.1 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.
64.2 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

64.3 Connecting to VMs via SSH Over Tor

64.3.1 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.

```bash
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:

```bash
$ 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 157xhqhl1t1u323avi.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)
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.

### 64.4 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.

### 65.1 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.

### 65.2 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.
65.3 Developer Workflow

65.3.1 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.

65.3.2 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
```

65.3.3 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. Your 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)
def check_upgrade(self):
    '''This function is run after the upgrade and verifies the state
    of the database or filesystem. It MUST raise an exception if the
    check fails.
    '''
    pass

class DowngradeTester:
    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 downgrade.
        '''
        pass
    def check_downgrade(self):
        '''This function is run after the downgrade and verifies the state
        of the database or filesystem. It MUST raise an exception if the
        check fails.
        '''
        pass

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 pos-
sible 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.

## 65.3.4 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 su`
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
CHAPTER 66

Internationalization (i18n)

The code, templates and JavaScript user visible strings must all be wrapped with gettext functions to be substituted with the equivalent localized string. For instance:

```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'))
```

The gettext function reads .mo files to find the translation for the string given in argument at runtime. It 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 ""
```

For each language to be translated, a directory is created such as securedrop/translations/fr_FR and populated with a .po file derived from securedrop/translations/messages.pot, for translators to work with. For instance securedrop/translations/fr_FR/LC_MESSAGES/messages.po is almost identical to securedrop/translations/messages.pot except for the msgstr field which contains the translation:

```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."
```

This file is compiled into an optimized binary form, the .mo file used by the gettext function at runtime.

The Weblate web application is used to translate strings and relies on the gettext format behind the scene. It owns the securedrop/translations/messages.pot file and all other translation related files. The SecureDrop code is modified by sending a pull request but the translations are exclusively modified via Weblate.

The desktop icon 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 repository. They are then installed when configuring Tails with the tasks/create_desktop_shortcuts.yml tasks.
66.1 The i18n_tool.py Translation Helpers

The pybabel and gettext command line is wrapped into the i18n_tool.py translate-messages and i18n_tool.py translate-desktop helpers for convenience. It is designed to be used by developers, to run tests with fixtures and for packaging.

66.2 Creating New Translations

A user with weblate admin rights must visit the Weblate translation creation page and the Weblate desktop translation creation page to add the desired languages.

66.3 Updating Strings to be Translated

After strings are modified in the code, templates, JavaScript or desktop labels, the securedrop/translations/messages.pot files must also be updated. Individual developers should NOT do this when changing strings in the code; the translations will be updated in bulk later on.

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 will update securedrop/translations/messages.pot, install_files/ansible-base/roles/tails-config/templates/desktop.pot, and the messages.po files for each language.

Note: The changes will only be visible in the Weblate web interface used by translators after Merging Develop into the Weblate Fork.

66.4 Compiling Translations

`gettext` needs a compiled file for each language (the *.mo files). This can be done by running the following command:

```
securedrop/bin/dev-shell ./i18n_tool.py --verbose translate-messages --compile
```

For desktop files the compilation phases creates a modified version of the original file which includes 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
```

66.5 Verifying Translations

After a translation is compiled, the web page in which it shows can be verified visually by navigating to the corresponding state from http://localhost:8080 for the source interface or http://localhost:8081 for the journalist interface after running the following:
An easier way is to generate screenshots for each desired language with:

```bash
$ securedrop/bin/dev-shell bash
$ export PAGE_LAYOUT_LOCALES=en_US,fr_FR
$ ./i18n_tool.py --verbose translate-messages --compile
$ pytest -v --page-layout tests/pageslayout
...
...TestJournalistLayout::test_col_no_documents[en_US] PASSED
...TestJournalistLayout::test_col_no_documents[fr_FR] PASSED
...
```

**Note:** if unset, PAGE_LAYOUT_LOCALES defaults to en_US.

The screenshots for fr_FR are available in securedrop/tests/pageslayout/screenshots/fr_FR and the name of the file can be found in the function that created it in securedrop/tests/pageslayout/test_journalist.py or securedrop/tests/pageslayout/test_source.py.

### 66.6 Merging Translations Back to Develop

Weblate automatically pushes the translations done via the web interface as a series of commit 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 to the develop branch via pull requests for merge on a regular basis.

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 supported languages is hard-coded in the i18n_tool.py file, in the SUPPORTED_LANGUAGES variable. When a new language is fully translated and reviewed, the i18n_tool.py file must be manually edited to add this new language to the SUPPORTED_LANGUAGES variable.

```bash
$ git clone https://github.com/freedomofpress/securedrop
$ cd securedrop
$ git checkout -b wip-i18n origin/develop
$ securedrop/bin/dev-shell ./i18n_tool.py --verbose update-from-weblate
$ securedrop/bin/dev-shell ./i18n_tool.py --verbose translate-desktop --compile
$ securedrop/bin/dev-shell ./i18n_tool.py --verbose update-docs
$ git commit -m 'l10n: compile desktop files' translations # if needed
$ git push wip-i18n # and make a pull request from the branch
```

**Warning:** It is **very** important to carefully check each translated string does not look strange. 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 reviewer will manipulate a translated string to introduce a vulnerability in SecureDrop. But it is easy to check visually and significantly reduce the risk.

List contributors for each supported language:

```bash
$ for l in $sm ; do echo -n "$l " ; git log --format=%aN lab/i18n -- install_files/...ansible-base/roles/tails-config/templates/$l.po securedrop/translations/$l/LC_MESSAGES/messages.po | sort -u | tr '\n' ',' | sed -e 's/,/, /g' ; echo ; done
nl Anne M, kwadronaut, Yarno Ritzen,
```

(continues on next page)
fr Alain-Olivier,
...

Verify the translations are not broken:

```
$ securedrop/bin/dev-shell ./i18n_tool.py --verbose translate-messages --compile
$ PAGE_LAYOUT_LOCALES=$(echo $sm | tr ' ' ',') \
    pytest -v --page-layout tests/pageslayout
```

Go to https://github.com/freedomofpress/securedrop and propose a pull request.

**Note:** contrary to the applications translations, the desktop 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.

---

### 66.7 Merging 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 extract new strings to translate or existing strings that were updated.

The translations must be suspended in Weblate to avoid conflicts.

- Go to the Weblate commit page for SecureDrop

- Click **Lock**
The develop branch can now be merged into i18n as follows:

```
$ git clone https://github.com/freedomofpress/securedrop
$ cd securedrop
$ git remote add lab git@github.com:freedomofpress/securedrop-i18n.git
$ git fetch lab
$ git checkout -b i18n lab/i18n
$ git merge origin/develop
$ make -C securedrop translate
```

The translate Makefile target relies on the `i18n_tool.py` command to examine all the source files, looking for strings that need to be translated (i.e. `gettext('translate me')` etc.) and update the `.pot` and `.po` files, removing, updating and inserting strings to keep them in sync with the sources. Carefully review the output of `git diff`. Check `messages.pot` first for updated strings, looking for formatting problems. Then review the `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:

```
$ git commit -a -m 'l10n: sync with upstream origin/develop'
$ git push lab i18n
```

- 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.

```
$ ssh debian@weblate.securedrop.org
$ cd /app/weblate
$ sudo docker-compose run weblate rebuild_index --all --clean
```

Note that the new index will not be used right away, some workers may still have the old index open. Rebooting the machine is an option, waiting for a few hours is another option.
66.9 Release Management

66.9.1 Two Weeks Before the Release: Update

The new and updated strings are uploaded to Weblate. This is done late in the SecureDrop release cycle so translators get less notifications. It would be inconvenient if there were hundreds of strings needing attention. But SecureDrop is small and it is ok to postpone notifications.

- 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 MM DD, YY midnight. String freeze will be in effect MM DD, YY midnight.*
- Create a pull request for every source string suggestion coming from translators
- Backport every commit changing a source string to the release branch
- Update the i18n timeline and Weblate whiteboard

66.9.2 One Week Before the Release: String Freeze

- Verify develop and the release branch have the same source strings
- Merge develop into the Weblate fork
- Post an announcement to the translation section of the forum (see an example)
- Remind all developers about the string freeze, in the chat room
- Add a prominent Weblate whiteboard announcement that reads *The X.Y.Z deadline is Month day, year midnight. String freeze is in effect: no source string are modified before the release.*
- Update the i18n timeline and Weblate whiteboard

66.9.3 The Day of the Release

- 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

66.10 Translator Credits

Verify the names and emails look ok, otherwise add to .mailmap until it does:

```
$ git clone https://github.com/freedomofpress/securedrop
$ cd securedrop
$ git remote add lab git@github.com:freedomofpress/securedrop-i18n.git
$ git fetch lab
$ previous_version=0.4.4
```

(continues on next page)
We do not want to publish the translator emails so we strip them:

```bash
$ git log --pretty='%aN <aE>' $previous_version..lab/i18n -- \
    securedrop/translations install_files/ansible-base/roles/tails-config/templates | \n    → sort -u
```

### 66.11 Translations Admins

**Note:** The privilege escalation workflow is different for *code maintainers* and *translation maintainers*.

A translation admin is a person who is actively performing administrative duties. They have special permissions on the repositories and the translation platform. When someone is willing to become an admin, a thread is started in the translation section of the forum. If there is a consensus, the permissions of the new admin are elevated after a week or more. If there is no consensus, a public vote is organized among the current admins.

All admins are listed in the forum introduction page

The privileges of an admin who has not been active for six months or more are revoked. 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 translators community has a different set of rules and permissions, reason why it makes sense to have an independent policy.

#### 66.11.1 Admin Permissions

An admin may not need or want all permissions but they are entitled to have all of them.

- https://weblate.securedrop.org/admin/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

#### 66.11.2 Granting Reviewer Privileges in Weblate

- visit https://weblate.securedrop.org/admin/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
Getting Started with Weblate Translations

SecureDrop is a whistleblower submission system for media organizations to securely accept documents from and communicate with anonymous sources. For more information, you may be interested to:

- watch The Globe and Mail guide to using SecureDrop
- read the Localizationlab Ask me Anything on SecureDrop
- learn about what makes SecureDrop unique.

SecureDrop uses the Weblate platform. This page lists detailed instructions for getting started with the Weblate platform as a translator. Before starting this guide, you should know which language you would like to translate.

### 67.1 Who Uses SecureDrop?

There are two kind 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 all received proper training and know what a PGP public key is, they understand the basic workflow of SecureDrop.

### 67.2 Register for Weblate

#### 67.2.1 Login via GitHub for the First Time

If you already have a GitHub account, go to the Weblate registration page:
Click on the octocat icon and agree to login via GitHub.
You will then be redirected to the Weblate dashboard.
67.2.2 Register a Weblate Account

Go to the Weblate registration page

By registering you agree to use your name and email in version control system commits and provide your contribution under license defined by each translated project.

Fill the form and click **Register** and check your email for a message with the subject [Weblate] Your registration on Weblate containing a confirmation link from admin@securedrop.org. Confirming by clicking the first link will redirect you to the Weblate dashboard:
67.3 Choose a Language

From your Weblate dashboard:

Choose your languages in preferences and you will get an overview here of available translations for those languages in your watched projects.

Click the Watched translations menu and select Suggested translations:
And click the **translate** button to the right on the line that shows your native language:

---

**67.4 Translate a Phrase**

Each translatable string is shown in a **Source** box and you can translate it right below in the **Translation** text area. When you are done, click **Save** and the next untranslated string will appear.
For most strings a contextual screenshot is available on the right side showing how it looks in the software. Click to see it in full size.
67.5 Instructions About a User Navigation

Strings also come with instructions, further explaining them, and detailing where they are to be found in the user interface of the software, in case you still aren’t sure. When doing so, also verify that the steps needed to find it are correct.:

...tap "Set up account"...

To find this, navigate to the localized interface of FreeOTP to get a firmer grasp of what it does, and then translate the string displayed in place of Set up account.

67.6 Placeholders

Strings may contain placeholders, seen as content in between braces: These pick in other names and functions of the software, and must be left unmodified, but they can be moved around in a string. For instance:
Edit user {user} will 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}

### 67.7 Reviews

Translated strings are not put into SecureDrop before they are also reviewed. 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 can. You can ask for your _translator_ status to be elevated to _reviewer_ by posting a message in the translation category of the SecureDrop forum.

A reviewer sees Waiting for review and Approved radio buttons next to strings, all of which are initially set to Waiting for review.

When the translation is deemed correct, the reviewer should change it to Approved.
When in this state, only reviewers can modify the string. Translators can still suggest modifications if they notice something wrong, and comment if they disagree.

67.8 Glossary

A glossary is available, explaining terms specific to SecureDrop. It is also important that key terms are understood and precisely translated.

67.8.1 Adversary

Your adversary is the person or organization attempting to undermine your security goals. Adversaries can be different, depending on the situation. For instance, you may worry about criminals spying on the network of a cafe, or your
classmates at a school. Often the adversary is hypothetical.
This definition is an edited version copied from the EFF glossary

67.8.2 Air gap

If there are no networked means of communicating with a computer you own, meaning a computer or a whole network is physically isolated from all other networks, including by its very nature the Internet, is said to be air-gapped.
This definition is an edited version copied from the EFF glossary

67.8.3 Attack

In terms of computer security, an attack is a method used to attempt compromising security, or just gaining access to its actual use. An attacker is the person or organization carrying out an attack. An attack method, something targeting a weakness in the security, is sometimes called an “exploit.”
This definition is an edited version copied from the EFF glossary

67.8.4 Command Line Tool (command)

The “command line” works by way of giving a computer a series of small, self-contained orders (think of those science fiction movies where teenage geniuses type long strings of green text onto black screens). To use a command line tool, the user types a command into a window called a terminal emulator, hits the return or enter key, and then receives a textual response in the same window. Windows, Linux|GNU and Apple desktop computers still let you run software using this interface, as is the case on some mobile phones if you install the right app. The command line can be used to run software pre-packaged with your operating system, or install new ones. Some downloadable programs, especially technical utilities, use the command line instead of a more familiar “icons and buttons” user interface. The command line needn’t be scary, but it does require you to type in exactly the right set of letters and numbers to get the correct result, and it’s often unclear what to do if the responses don’t match your expectations.
This definition is an edited version copied from the EFF glossary

67.8.5 Cryptography

The art of designing secret codes or ciphers that let you exchange messages with a recipient without others being able to understand the message.
This definition is an edited version copied from the EFF glossary

67.8.6 Decrypt

Make a secret message or data intelligible. The idea behind encryption is to make messages that can only be decrypted by the person or people meant to receive them.
This definition is an edited version copied from the EFF glossary

67.8.7 Encryption

A process that takes a message and makes it unreadable except to a person who knows how to decrypt it back into a readable form.
This definition was copied from the EFF glossary
67.8.8 Encryption Key

An encryption key is a piece of information that is used to convert a message into an unreadable form. In some cases, you need the same encryption key to decode the message. In others, the encryption key and decryption key are different.

This definition was copied from the EFF glossary

67.8.9 Fingerprint

The keys of public key cryptography are very large numbers, sometimes a thousand or more digits long. A fingerprint is a much smaller number or set of numbers and letters that can be used as a unique name for that key, without having to list all of the key’s digits. So, for instance, if you and a friend wished to make sure you both had the same key, you could either spend a long time reading off all the hundreds of digits in the key, or you could each calculate your key’s fingerprint and compare those instead. The fingerprints presented by cryptographic software usually consist of around 40 letters and numbers. If you carefully check that a fingerprint has the right value, you should be safe against impersonation using a fake key. Some software tools may offer more convenient alternative ways to verify a friend’s key, but some form of verification needs to happen to prevent communications providers from easily being able to listen in.

This definition was copied from the EFF glossary

67.8.10 HTTPS

If you’ve ever seen a web address spelled out as “http://www.example.com/”, you’ll recognize the “http” bit of this term. HTTP (hypertext transfer protocol) is the way a web browser on your machine talks to a remote web server. Unfortunately, standard http sends text insecurely across the Internet. HTTPS (the S stands for “secure”) uses encryption to better protect the data you send to websites, and the information they return to you, from prying eyes.

This definition was copied from the EFF glossary

67.8.11 Key

In cryptography, a piece of data which gives you the capability to encrypt or decrypt a message.

This definition was copied from the EFF glossary

67.8.12 Keyring

If you use public key cryptography, you’ll need to keep track of many keys: your secret, private key, your public key, and the public keys of everyone you communicate with. The collection of these keys is often referred to as your keyring.

This definition was copied from the EFF glossary

67.8.13 Man-in-the-Middle Attack (MITM)

Suppose you believe you were speaking to your friend, Bahram, via encrypted instant messager. To check it’s really him, you ask him to tell you the city where you first met. “Istanbul” comes the reply. That’s correct! Unfortunately, without you or Bahram knowing, someone else online has been intercepting all your communications. When you first connected to Bahram, you actually connected to this person, and she, in turn, connected to Bahram. When you think you are asking Bahram a question, she receives your message, relays the question to Bahram, receives his answer back
, and then sends it to you. Even though you think you are communicating securely with Bahram, you are, in fact, only communicating securely with the spy, who is also communicating securely to Bahram! This is the man-in-the-middle attack. Men-in-the-middle can spy on communications or even insert false or misleading messages into your communications. Security-focused internet communications software needs to defend against the man-in-the-middle attack to be safe against attackers who have control of any part of the Internet between two communicators.

This definition was copied from the EFF glossary

### 67.8.14 Public Key Encryption

Traditional encryption systems use the same secret, or key, to encrypt and decrypt a message. So if I encrypted a file with the password “bluetonicmonster”, you would need both the file and the secret “bluetonicmonster” to decode it. Public key encryption uses two keys: one to encrypt, and another to decrypt. This has all kinds of useful consequences. For one, it means that you can hand out the key to encrypt messages to you, and as long as you keep the other key secret, anyone with that key can talk to you securely. The key you hand out widely is known as the “public key”: hence the name of the technique. Public key encryption is used to encrypt email and files by Pretty Good Privacy (PGP), OTR for instant messaging, and SSL/TLS for web browsing.

This definition was copied from the EFF glossary

### 67.8.15 Two-Factor Authentication

“Something you know, and something you have.” Login systems that require only a username and password risk being broken when someone else can obtain (or guess) those pieces of information. Services that offer two-factor authentication also require you to provide a separate confirmation that you are who you say you are. The second factor could be a one-off secret code, a number generated by a program running on a mobile device, or a device that you carry and that you can use to confirm who you are. Companies like banks, and major internet services like Google, PayPal and Twitter now offer two-factor authentication.

This definition is an edited version copied from the EFF glossary

### 67.9 Weblate Glossary

For each string to be translated, Weblate shows a glossary of terms and their translation to help unify their translations. For instance when translating Please wait for a new two-factor token before logging in again, Weblate notices the word two-factor is found in the glossary and displays the translation in the glossary to the right.
Before translating strings, it is recommended to add all terms in the SecureDrop localization glossary by clicking on the pen in the right corner of the glossary displayed with each translated string and then Add new word:

When all the terms are in the glossary, it is recommended to take another look at the full list of terms and verify there is no duplicate or other mistakes.
Tip: The terms copied from the EFF glossary already have a translation in a number of languages.

67.10 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

67.11 Frequently Asked Questions

- What if the language I want to translate is not on the list?
  
  You can send a request for a new language in the translation category of the SecureDrop forum. But please make sure the language you want is not already present.
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.

### 68.1 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.

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.
68.2 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.

68.3 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.

68.4 Style Guide

68.4.1 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.)

68.4.2 Terminology

Text taken directly from a user interface is in **bold face**.

“Once you’re sure you have the right drive, click **Format Drive**.”

SecureDrop-specific **terminology** 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

68.4.3 Code Blocks

Ensure that example commands in codeblocks are easy to copy and paste. Do not prepend the `$` shell prompt indicator to example commands:
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
```

### 68.4.4 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`.

### 68.4.5 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.
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.

### 70.1 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.

### 70.2 Running the Application Tests

The tests can be run inside the development VM:

```
make -C securedrop test
```

Or the app-staging VM:
```bash
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.

### 70.2.1 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
```

### 70.3 Updating the Application Tests

Unit tests are stored in the `securedrop/tests/` directory and functional tests are stored in the functional test directory:

```bash
securedrop/tests/
    functional
    ├── test_admin_interface.py
    ├── test_submit_and_retrieve_file.py
    │   ├── submission_not_in_memory.py
    │   └── ...
    └── utils
        ├── db_helper.py
        ├── env.py
        └── async.py
    └── test_journalist.py
    └── test_source.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 spectest.

### 71.1 Installation

```bash
pip install -r securedrop/requirements/develop-requirements.txt
```

### 71.2 Running the Config Tests

In order to run the tests, first create and provision the VM you intend to test.

For the staging VMs:

```bash
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:

#### 71.2.1 libvirt:

```bash
molecule verify -s libvirt-staging
```
71.2.2 virtualbox:

```bash
molecule verify -s virtualbox-staging
```

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`.

71.3 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:

```bash
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:

```bash
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.

71.4 Config Test Layout

With some exceptions, the config tests are broken up according to platform definitions in the Molecule configuration:
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.

## 71.5 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.

## 72.1 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.
...
```

72.1.1 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.

72.1.2 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.

74.1 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.

74.2 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.

## 74.3 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.

   **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.

**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.
74.4 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

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.

```
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.

### 76.1 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.
Release Management

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.

### 77.1 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`:

   ```
   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.

### 77.2 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:

```bash
git tag -d 0.x.y
```

6. Make the signed tag:

```bash
git mktag < 0.x.y.tag > .git/refs/tags/0.x.y
```

7. Verify the signed tag:

```bash
git tag -v 0.x.y
```

8. Push the signed tag:

```bash
git push origin 0.x.y
```

9. Build Debian packages. People building Debian packages should verify and build off the signed tag.

10. Step through the signing ceremony for the Release file(s) (there may be multiple if Tor is also updated along with the SecureDrop release).


12. Coordinate with one or more team members to confirm a successful clean install in production VMs using the packages on `apt-test.freedom.press`.

13. Put signed Debian packages on `apt.freedom.press`. The release is now live.

14. Make sure that the default branch of documentation is being built off the tip of the release branch.

15. Make sure that release notes are written and posted on the SecureDrop blog.

16. Make sure that the release is announced from the SecureDrop Twitter account.

### 77.3 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).
78.1 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

78.2 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.
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.

79.1 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-trusty-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 Trusty.

79.2 Download Ubuntu Trusty server ISO

On sd-dev, download the Ubuntu Trusty 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.

79.3 Create the Trusty 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.

### 79.4 Boot into installation media

In dom0:

```
qvm-start sd-trusty-base --cdrom=sd-dev:/home/user/ubuntu-14.04.5-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-trusty-base ip`, with /24 netmask suffix
- **Gateway**: use value returned by `qvm-prefs sd-trusty-base visible_gateway`
- **For DNS**, use Qubes’s DNS servers: 10.139.1.1 and 10.139.1.2.
- **Hostname**: `sd-trusty-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-trusty-base
```

in dom0. You should get a login prompt.

### 79.5 Initial VM configuration

Before cloning this machine, we’ll update software to reduce provisioning time on the staging VMs. In the new `sd-trusty-base` VM’s console, do:

```
sudo apt update
sudo 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
```

When initial configuration is done, run `qvm-shutdown sd-trusty-base` to shut it down.
79.6 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-trusty-base sd-app-base
qvm-clone sd-trusty-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.

79.6.1 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.

Finally, 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.

79.6.2 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:

```
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
```

(continues on next page)
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

```
sudo sh /rw/config/qubes-firewall-user-script
```

Now from sd-dev, you should be able to do

```
ssh sdadmin@10.137.0.50
```

and log in with the password securedrop.

### 79.6.3 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:

```
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.

### 79.7 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](https://secure-drop-documentation.readthedocs.io/en/latest/) to set up the development environment.

Once finished, build the Debian packages for installation on the staging VMs.

```
make build-debs
```

The .deb files will be available in `build/`.

### 79.8 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:
79.9 Creating staging instance

After creating the Standalone VMs as described above:

- sd-dev
- sd-trusty-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:

```
molecule test -s qubes-staging
```

**Note:** The reboot actions run against the VMs during provisioning will only shutdown the VMs, not start them again, since these are Standalone VMs. Therefore the test action will fail by default, unless you judiciously run `qvm-start <vm>` for each VM after they’ve shut down.

You can use the smaller constituent Molecule actions, rather than the bundled test action:

```
molecule create -s qubes-staging
molecule converge -s qubes-staging
```

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.
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.

### 80.1 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 bento/ubuntu-16.04 # choose the virtualbox option
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:

#### 80.1.1 libvirt:

```bash
molecule login -s libvirt-staging-xenial -h app-staging
```
80.1.2 virtualbox:

```
molecule login --s virtualbox-staging-xenial -h app-staging
```

To run the testinfra tests against the Xenial environment, you can use the commands:

80.1.3 libvirt:

```
molecule verify --s libvirt-staging-xenial
```

80.1.4 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.

80.2 Known bugs with Xenial support

Below is a high-level overview of known problems to be addressed for delivering Xenial compatibility.

**Packaging** Dependencies for the `securedrop-app-code` deb package have changed; `apache2` should be explicitly required; `apache2-mpm-worker` should be omitted.

**Firewall** The `_apt` user should be permitted to perform DNS and outbound TCP calls on ports 80 and 443, rather than the `root` user.

**AppArmor** Explicit rules required for Apache mpm worker/event changes. `gpg2` policy should permit links via `/var/lib/securedrop/keys/*` or similar.

**PAM logic** The PAM common-auth customizations include declarations for `pam_ecryptfs.so` which prove problematic; commenting out ostensibly resolves. More research required.

**Config tests** The testinfra config test suite runs slightly different checks for Trusty and Xenial where appropriate. Care should be taken to preserve functionality of the config tests against both distros.

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
81.1 Assumptions

81.1.1 Assumptions About the Source

- The source acts reasonably and in good faith, e.g. if the user were to give their codename 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 or the Tor Browser.
- The source follows our guidelines for using SecureDrop.
- The source is accessing an authentic SecureDrop site.

81.1.2 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.
81.1.3 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.

81.1.4 Assumptions About the Source's Computer

- The computer correctly executes Tails or the Tor Browser.
- The computer is not compromised by malware.

81.1.5 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.

81.1.6 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.

81.1.7 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.

81.1.8 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.
81.1.9 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.

81.2 Attack Scenarios

81.2.1 What the Application Server Can Achieve

- The server sees the plaintext codename, used as the login identifier, of every source.
- 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 hashes of codenames, created with scrypt and randomly-generated salts.
- 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.

81.2.2 What the Monitor Server Can Achieve

- 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.
81.2.3 What the Workstations Can Achieve

- 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* and the journalist’s personal GPG key.

- 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.

81.2.4 What a Compromise of the Source’s Property Can Achieve

- 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.

81.2.5 What a Physical Seizure of the Source’s Property Can Achieve

- 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.

81.2.6 What a Compromise of the Admin’s Property Can Achieve

• 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 key.
• 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.

81.2.7 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.

81.2.8 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. 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.
  – Access the journalist’s personal GPG key.

• 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.

81.2.9 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.

81.2.10 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.

81.2.11 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. 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.
81.2.12 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:
  - ???

- 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.

81.2.13 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.

81.2.14 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 is in use at the time of compromise or seizure. A physical seizure of this machine, the Tails device or the Transfer device will also achieve nothing, assuming that Tails’ implementation of full-disk encryption works 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 GPG private key used to encrypt and decrypt submitted documents.
  - View, modify, and delete encrypted--and possibly also decrypted submissions--if the Transfer device is in use.

81.2.15 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 is in use at the time of compromise or seizure. A physical seizure of this machine, the Tails device or the Transfer device will also achieve nothing, assuming that Tails’ implementation of full-disk encryption works 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 GPG private key used to encrypt and decrypt submitted documents.
– View, modify, and delete encrypted—and possibly also decrypted submissions—if the Transfer device is in use.

81.2.16 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.

81.2.17 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.

81.2.18 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.