ArchiveBox / **ArchiveBox** Public

Code Issues 162 Pull requests 3 Discussions Actions Projects 1

Security Overview

Jump to bottom

Nick Sweeting edited this page 5 days ago · 71 revisions

> 💬 *We offer consulting services to set up, secure, and maintain ArchiveBox on your preferred hosting environment.*
>
> We use this revenue (from corporate clients who can afford to pay) to support open source development and keep ArchiveBox free.

Web UI Permissions

```
archivebox config --set PUBLIC_INDEX=False      # require login to access the li  f
archivebox config --set PUBLIC_SNAPSHOTS=False  # require login to access Snapshot co
archivebox config --set PUBLIC_ADD_VIEW=False   # require log-in to submit new URLs f

archivebox manage [createsuperuser|changepassword] # create/modify admin UI users
```

See Setting Up Authentication for more...

ArchiveBox Use-Cases

Archiving Public Content Only ⭐ `[Default, recommended for most people]` 🚪

This is the default (lax) mode, intended for archiving public (non-secret) URLs without authenticating the headless browser. This is the mode used if you're archiving news articles, audio, video, etc. browser bookmarks to a folder published on your webserver. This allows you to access and link to content on `http://your.archive.com/archive...` after the originals go down.

The default mode should not be used for archiving entire browser history or authenticated private content like Google Docs, paywalled content, invite-only subreddits, private photo share urls, etc.

```
# (these are the defaults)
archivebox config --set SAVE_ARCHIVE_DOT_ORG=True
archivebox config --set CHROME_USER_DATA_DIR=None
archivebox config --set COOKIES_FILE=None
```

Archiving Content Behind Log-Ins 🚨 `[Advanced users only]`

ArchiveBox is able to archive content that requires authentication or cookies, but it comes with some caveats. Create dedicated logins for archiving to access paywalled content, private forums, LAN-only content, etc. then share them with ArchiveBox via Chrome profile + cookies.txt file.

```
archivebox config --set SAVE_ARCHIVE_DOT_ORG=False
archivebox config --set CHROME_USER_DATA_DIR=/path/to/chrome/profile
archivebox config --set COOKIES_FILE=/path/to/cookies.txt
```

To get started, set `CHROME_USER_DATA_DIR` and `COOKIES_FILE` to point to a Chrome user folder that has your sessions and a wget `cookies.txt` file respectively.

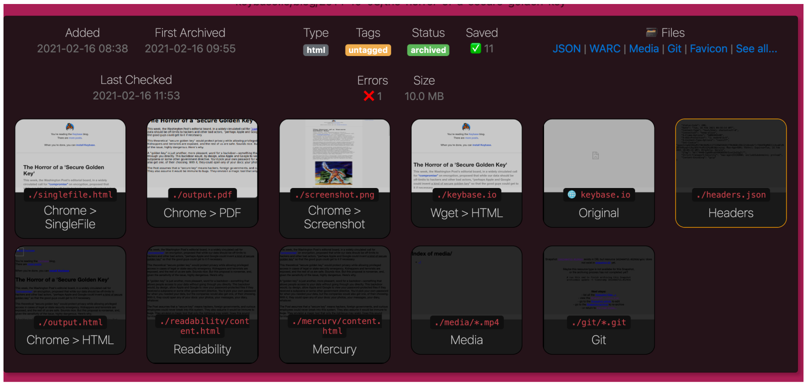
➡️ For full instructions on setting up a Chromium user profile see here: https://github.com/ArchiveBox/ArchiveBox/wiki/Chromium-Install#setting-up-a-chromium-user-profile

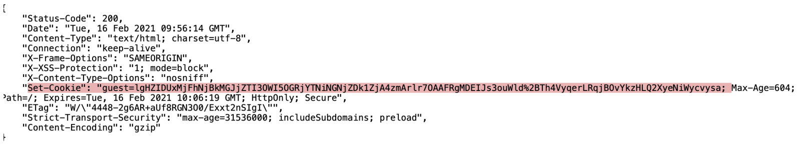
If you're importing private links or authenticated content, you probably don't want to share your archive folder publicly on a webserver, so don't follow the Publishing Your Archive instructions unless you are only serving it on a trusted LAN or have some sort of authentication in front of it. Make sure to point ArchiveBox to an output folder with conservative permissions, as it may contain archived content with secret session tokens or pieces of your user data. You may also wish to encrypt the archive using an encrypted disk image or filesystem like ZFS as it will contain all requests and response data, including session keys, user data, usernames, etc.

⚠️ Things to watch out for: ⚠️

- any cookies / secret state present in a Chrome user profile or `cookies.txt` file may be reflected in server responses and saved in the Snapshot output (e.g. in `headers.json`) making it visible in cleartext to anyone viewing the Snapshot, (don't use your personal Chrome profile for archiving or people viewing your archive can then authenticate as you!)
- any secret tokens embedded in URLs (e.g. secret invite links, Google Doc URLs, etc.) will be visible on `archive.org` as the URLs are not filtered when saving to `archive.org` (disable submitting to Archive.org entirely with `SAVE_ARCHIVE_DOT_ORG=False`)

- the domain portion in archived URLs is sent to a favicon service in order to retrieve an icon more reliably than a janky internal implementation would be able to (if leaking domains is a concern, you can change the `FAVICON_PROVIDER` or disable favicon fetching entirely with `SAVE_FAVICON=False`)
- viewing malicious archived JS could allow an attacker to access your other archive items + the admin interface (JS executes on the same origin as the admin panel right now, fix is pending, set `SAVE_WGET=False SAVE_DOM=False` to disable the risky extractors entirely or avoid viewing their output directly in a browser)





```
{
    "Status-Code": 200,
    "Date": "Tue, 16 Feb 2021 09:56:14 GMT",
    "Content-Type": "text/html; charset=utf-8",
    "Connection": "keep-alive",
    "X-Frame-Options": "SAMEORIGIN",
    "X-XSS-Protection": "1; mode=block",
    "X-Content-Type-Options": "nosniff",
    "Set-Cookie": "guest=lgHZIDUxMjFhNjBkMGJjZTI3OWI5OGRjYTNiNGNjZDk1ZjA4zmArlr7OAAFRgMDEIJs3ouWld%2BTh4VyqerLRqjBOvYkzHLQ2XyeNiWycvysa; Max-Age=604;
Path=/; Expires=Tue, 16 Feb 2021 10:06:19 GMT; HttpOnly; Secure",
    "ETag": "W/\"4448-2g6AR+aUf8RGN3O0/Exxt2nSIgI\"",
    "Strict-Transport-Security": "max-age=31536000; includeSubdomains; preload",
    "Content-Encoding": "gzip"
}
```

An example of a session cookie reflected in `headers.json` *visible in the archive.*

Publishing

> ⓘ Caution
>
> Re-hosting untrusted archived content on a domain can potentially compromise *all apps on that domain*!
> (including other subdomains)

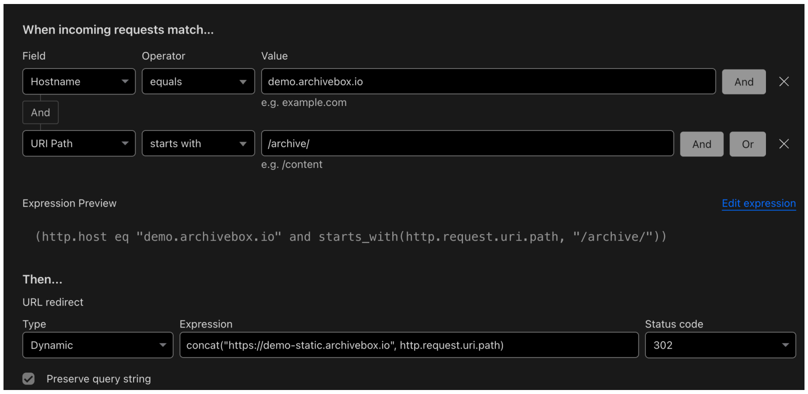
Make sure you thoroughly understand the dangers of hosting untrusted HTML/JS/CSS that may be captured during archiving, and how viewing it can enable CSRF attacks across all apps on the same domain. If a logged-in user happens to visit an archived page with malicious Javascript embedded, it would allow the JS to hijack any cookies on the domain and pretend to be them, potentially exfiltrating or modifying other Snapshots/data on your server.

(This is why we don't support serving ArchiveBox from a subdirectory like `myapps.example.com/archivebox/`, it's too dangerous to share domains)

The industry standard approach is to use a separate domain for untrusted content, for example Github uses `githubusercontent.com` and Google uses `googleusercontent.com` for all user-uploaded files. If hosting ArchiveBox publicly, do the same and keep it on an isolated domain in order to mitigate potential damage of leaked cookies, CORS, and CSRF attacks.

To protect the Admin dashboard, it's also recommended to serve all content under `/archive/` on a separate domain from `/admin/`. We do this on our servers using a simple redirect rule in nginx/cloudflare like so:

- https://demo.archivebox.io: only serves `/`, redirects `/archive/*` to `demo-static.`
- https://demo-static.archivebox.io: only serves `/archive/`, redirects everything else to `demo.`



When incoming requests match...

Field	Operator	Value
Hostname	equals	demo.archivebox.io
And		
URI Path	starts with	/archive/

Expression Preview

```
(http.host eq "demo.archivebox.io" and starts_with(http.request.uri.path, "/archive/"))
```

Then...

URL redirect

Type	Expression	Status code
Dynamic	concat("https://demo-static.archivebox.io", http.request.uri.path)	302

☑ Preserve query string

Published archives automatically include a `robots.txt` `Dissallow: /` to block search engines from indexing them. You may still wish to publish your contact info in the index footer though using `FOOTER_INFO` so that you can respond to any DMCA and copyright takedown notices if you accidentally rehost copyrighted content.

⚠️ Make sure to read all the warnings above about the dangers of exposing Chrome profile data, cookies, secret tokens in URLs, and the risks of viewing archived JS on a shared origin before publishing your archive.

More info:

- https://github.com/ArchiveBox/ArchiveBox/wiki/Publishing-Your-Archive
- https://github.com/ArchiveBox/ArchiveBox/wiki/Publishing-Your-Archive#security-concerns
- https://github.com/ArchiveBox/ArchiveBox/wiki/Publishing-Your-Archive#copyright-concerns
- https://en.wikipedia.org/wiki/Cross-site_request_forgery
- https://github.com/ArchiveBox/ArchiveBox/issues/239

Do not run as root

I SEE YOU LIKE TO RUN AS THE ROOT USER
I TOO LIKE TO LIVE DANGEROUSLY

> ⚠ Warning
>
> **Did you run a command in Docker with `exec` instead of `run` by accident and end up here?**
> Make sure you use `docker run` instead of `docker exec` to run ArchiveBox commands.
>
> *For example:*
> ✅ `docker compose run archivebox manage createsuperuser`
> ✅ `docker run -it -v $PWD:/data archivebox/archivebox manage createsuperuser`
> (`docker run` automatically uses the correct `archivebox` user & file permissions enforced via `./bin/docker_entrypoint.sh`)
>
> *instead of:*
> ❌ `docker compose exec archivebox manage createsuperuser`
> ❌ `docker exec -it archivebox manage createsuperuser`
> (`docker exec` will skip the entrypoint and attempt to run everything as root and fail)
>
> If you must use `exec` for some reason (e.g. if you only have access to a live container shell), you can run `su archivebox` within the shell, or add the arg `--user=archivebox` after `exec`.

Do not run ArchiveBox as root for a number of reasons:

- Chrome will execute as root and fail immediately because Chrome sandboxing is pointless when the data directory is opened as root (do not set `CHROME_SANDBOX=False` just to bypass that error!)
- All dependencies will be run as root, if any of them have a vulnerability that's exploited by sites you're archiving you're opening yourself up to full system compromise
- ArchiveBox does lots of HTML parsing, filesystem access, and shell command execution. A bug in any one of those subsystems could potentially lead to deleted/damaged data on your hard drive, or full system compromise unless restricted to a user that only has permissions to access the directories needed
- Do you really trust a project created by a Github user called `@pirate` 😉? Why give a random program off the internet root access to your entire system? (I don't have malicious intent, I'm just saying in principle you should not be running random Github projects as root)

Instead, you should run ArchiveBox under a separate user account with less privileged access:

```
useradd -r -g archivebox -G audio,video archivebox  # the audio & video groups a s
mkdir -p /home/archivebox/data
chown -R archivebox:archivebox /home/archivebox
...
sudo -u archivebox archivebox add ...
```

~~If you absolutely must run it as root for some reason, a footgun is provided: you can set `ALLOW_ROOT=True` via environment variable or in your ArchiveBox.conf file.~~ This footgun option was removed (I'm sorry, the support burden of helping people who messed up their systems by running everything as root was too high).

Output Folder

Database

The ArchiveBox database is an unencrypted, uncompressed SQLite3 `index.sqlite3` file on disk, and such does not require an authenticated admin SQL login to access (like PostgreSQL/MySQL would). Make sure to protect your database file adequately as anyone who can read it can read your entire collection contents. Passwords for the admin users are stored as salted and PBKDF2 hashed strings in the `auth_user` table.

More info:

- https://github.com/ArchiveBox/ArchiveBox/wiki/Usage#disk-layout
- https://github.com/ArchiveBox/ArchiveBox/wiki/Upgrading-or-Merging-Archives#database-troubleshooting
- https://github.com/ArchiveBox/ArchiveBox/wiki/Upgrading-or-Merging-Archives#modify-the-archivebox-sqlite3-db-directly
- https://github.com/ArchiveBox/ArchiveBox/wiki/Upgrading-or-Merging-Archives#example-adding-a-new-user-with-a-hashed-password

Filesystem

How much are you planning to archive? Only a few bookmarked articles, or thousands of pages of browsing history a day? If it's only 1-50 pages a day, you can probably just stick it in a normal folder on your hard drive, but if you want to go over 100 pages a day, you will likely want to put your archive on a compressed/deduplicated/encrypted disk image or filesystem like ZFS. Other distributed/networked/checksummed filesystems that have also been reported to work (but are not technically officially supported) include SMB, NFS, Ceph, Unraid, and BTRFS. Make sure the filesystem you're using supports FSYNC. Some filesystems are unable to store more than a certain number of directory entries, and your total number of snapshots in `./archive` may be capped as a result. Some other filesystems begin to have performance degradations but continue to function when the directory entry count gets too high. Generally this isn't an issue unless you have more than ~20,000 Snapshot folders in `./archive`.

Purging entries

Unless `--yes --delete` is passed to `archivebox remove`, Snapshots removed from the index remain in the filesystem and their `./archive/<timestamp>` folders need to be deleted manually to be fully removed. Imported URLs are also logged separately in `./sources`, `./logs`, and the Sonic full-text index `./sonic` and should be removed manually as well to clear all traces of a URL added by accident. You can search for a URL on the filesystem you're trying to remove using `grep -a -r "https://example.com/url/to/search/for"`.

Permissions

Consider what permissioning to apply to your archive folder carefully. Limit access to the fewest possible users by checking folder ownership and setting `OUTPUT_PERMISSIONS` accordingly. Generally the `index.sqlite3` file, `archive/` folder, and `ArchiveBox.conf` file must all be owned and writable by the `archivebox` user or a dedicated non-root user.

`PUID` & `PGID` can be set when running with Docker to control what user and group ArchiveBox expects to own the data directory within the container.

More info:

- https://github.com/ArchiveBox/ArchiveBox/wiki/Usage#disk-layout
- https://github.com/ArchiveBox/ArchiveBox#output-formats
- https://github.com/ArchiveBox/ArchiveBox/wiki/Upgrading-or-Merging-Archives#database-troubleshooting
- https://github.com/ArchiveBox/ArchiveBox/wiki/Upgrading-or-Merging-Archives#filesystem-doesnt-support-fsync-eg-network-mounts
- https://github.com/ArchiveBox/ArchiveBox#storage-requirements

✏️ Help improve our documentation...

▸ **Pages** 37

Getting Started

- 🔢 Quickstart
- 🖥️ Install
- 🐳 Docker
- ➡️ Supported Sources
- ⬅️ Supported Outputs

Usage

- $ Command Line
- 🌐 Web UI
- 🧩 Browser Extension
- 👾 REST API / Webhooks
- 📜 Python API / REPL / SQL API

Reference

- ⚙️ Configuration
- 📦 Dependencies

- 💿 Disk Layout
- 🔒 Security Overview
- 📝 Developer Documentation

Guides

- Upgrading
- Setting up Storage (NFS/SMB/S3/etc)
- Setting up Authentication (SSO/LDAP/etc)
- Setting up Search (rg/sonic/etc)
- Scheduled Archiving
- Publishing Your Archive
- Chromium Install
- Cookies & Sessions Setup
- Merging Collections
- Troubleshooting

More Info

- ⭐ Web Archiving Community
- Background & Motivation
- Comparison to Other Tools
- Changelog & Roadmap

Stars 21k Donate Directly

Github Sponsors Patreon

Community Chat Forum Zulip

Clone this wiki locally

```
https://github.com/ArchiveBox/ArchiveBox.wiki.git
```