How GCHQ made its CyberChef app open source

CyberChef open source app statistics

CyberChef is a web app to carry out many cyber operations within a web browser. It has over 300 operations, including basic encoding with Base64, Advanced Encryption Standard (AES) decryption, or changing character encodings. The app can handle many operations at once, making it a quick way to experiment and translate data.

The CyberChef app:

- was created in an analyst's 10% 'innovation time'
- had its code fully opened in November 2016 under the Apache 2.0 license
- has been used in almost every country around the world
- has 75,000+ users per month
- has had 1.4 million unique users since its release
- has 75 open source community contributors

About GCHQ and CyberChef

GCHQ is the UK's intelligence, security and cyber agency with a mission to help keep the UK safe. Its people use cutting-edge technology, technical ingenuity and world-leading partnerships to identify, analyse and disrupt threats in an increasingly complex world.

GCHQ believes that success depends on great minds not thinking alike. To promote innovation and personal development alongside their usual work, staff are encouraged to spend 10% of their hours on projects of their choosing related to their work. In 2014, an analyst began looking for open tools to help automate some data manipulation operations. It was too time consuming to write short scripts for every data transformation they needed such as encoding, encryption and viewing data in different formats.

Seeing few open source tools available, they began to develop what would later become known as CyberChef.

Understanding the CyberChef app's design

The CyberChef web app has a simple design and includes:

- an input box to add the file or text you want to work with
- a list of all the operations you can run on the file or text
- a 'recipe' box where you can drag the operations you want to use and specify how you want to use them
- an output box to display the results of your operations

CyberChef is 100% client-side. No input or information about the combination

of operations you use (known as ‘recipes’ in the app) is sent outside your browser.

GDS interviewed the anonymous creator of CyberChef, who said: “We understand that in the cybersecurity industry, people are often working on data that they want to keep to themselves due to commercial or personal sensitivities, so running all the processing locally is very important to us.”

CyberChef:

- is built in JavaScript
- uses webpack to generate bundles
- uses Grunt as a build system
- uses GitHub Actions for continuous integration
- is hosted with GitHub Pages
- uses ESLint for linting and has a test suite written by external contributors
- uses Nightwatch.js to test the user interface
- is also available via npm

Considering making CyberChef open source

After building the app for their own use, the creator began sharing CyberChef with:

- their colleagues in GCHQ
- other UK agencies and partners
- international government intelligence agencies

CyberChef became so well-used that other analysts started asking if the app could be shared more widely with industry, students, businesses, and anyone who wanted to try translating data. So the creator began to explore the possibility of opening CyberChef to the public.

Why GCHQ decided to open the code

The creator knew opening the CyberChef code would:

- provide a useful app for the wider industry
- help users to suggest bug fixes, contributions and ideas
- encourage others to experiment with data, encoding, encryption and computer science

GCHQ also wanted to be as transparent as it could while protecting operational secrets, encouraging interest in cyber security, and demonstrating its support of the open source community.

As its creator explained: “GCHQ has put a lot of effort into increasing transparency, so it makes sense that, where possible, we share apps like CyberChef, so everyone can use it. It helps to demystify what we’re doing a little and build trust.”

However, publicly releasing a cyber security app from a world-leading intelligence agency would require careful planning and execution. The creator knew they needed to:

- get appropriate approvals from GCHQ
- assess the possible risks of opening the app and its code
- mitigate any risks appropriately
- decide how to publish CyberChef's code in the open
- agree how the app would be maintained and how they would manage contributions from non-GCHQ staff

Getting approval for CyberChef

CyberChef was not the first product GCHQ had released to the public. In 2015, the agency opened a graph database framework called Gaffer.

Knowing GCHQ had approved open code before, the creator began speaking with relevant teams such as the Innovation team and the Legal and Policy team to investigate what approvals were needed. The reaction was overwhelmingly supportive.

"To begin with I was really concerned about whether we would be able to make this an open source app due to the nature of our work, but I was put at ease by colleagues who made sure there was nothing sensitive being released," says the CyberChef creator. After speaking with a few boards and departments to make sure they would not reveal anything sensitive, the creator moved to the technical process of opening the code.

Choosing a license

The creator wanted to make CyberChef a fully-fledged open source product, rather than simply publishing the code online. When releasing the app under an open source license, GCHQ would remain a major contributor. This meant they would manage control over all contributions and encourage people to use the app with appropriate credit given, but the app would not 'belong' to the organisation anymore.

GCHQ needed to choose an appropriate license to encourage the open source community to use and contribute to CyberChef while users had to give GCHQ credit where it was due. They chose Apache 2.0.

Some existing code language libraries were not compatible with Apache 2.0, but after a few changes and library substitutions, the code was ready to release.

Opening the code

When it came to publishing the code, the CyberChef creator chose GitHub as it was considered the industry standard for open source software hosting.

After getting the necessary sign-off, the creator:

1. Reviewed the code on GCHQ's private networks to make sure it was presentable, readable, clear of any personal or sensitive data, and safe to release.
2. Tested the new open-friendly code on a device which had access to the public internet.
3. Practised some scenarios of contributing to and publishing the code in the open.
4. Published the code in a new repository on GitHub.

Maintaining anonymity

From working at GCHQ, the CyberChef creator needed to maintain anonymity but in a way which complemented open collaboration. They decided to use a string of random numbers as their username.

Deciding how to manage contributions

One of the main benefits of opening code is the ability to accept external contributions. Alongside the Apache 2.0 license, the creator published a contributor agreement. The GCHQ OSS Contributor License Agreement explains what constitutes a contribution, and the ownership and intellectual property rights of users when making a contribution to CyberChef.

CyberChef has received many interesting contributions. An example from early on in the project is how one contributor implemented a test suite for CyberChef's operations. The CyberChef team still uses this test suite today.

CyberChef has a wiki with some code conventions and design principles for people contributing to the project. These code conventions and design principles are kept deliberately broad to avoid being too prescriptive, as the creator wants to minimise barriers for people contributing.

CyberChef maintains the right to refuse contributions but so far, the quality of contributions has been high. CyberChef has a linter and test suite built into the build process and if these flag contributions, people can usually fix the contributions themselves.

Managing CyberChef

GCHQ employees voluntarily manage the CyberChef app alongside their day-to-day work.

Managing reviews and version changes

For versioning, CyberChef uses semver for 3 different levels of changes.

1. Patch changes for bug fixes or small tweaks.
2. Minor changes for when CyberChef adds new features or operations.
3. Major changes for when CyberChef introduces a major new feature or restructure, for example the 'Magic' operation.

Communicating with users

GCHQ communicates with users of CyberChef by:

Impact of opening CyberChef

GCHQ has used CyberChef for educational and awareness programmes. For example, the competition CyberFirst Girls Competitions exists for year 8 schoolgirls to increase their awareness and interest in cybersecurity and computer science. It encourages participants to use CyberChef to solve some of the challenges in the competition.

Lessons learned in opening CyberChef

“Just go for it,” is CyberChef’s creator’s advice.

Managing and getting stakeholder buy-in was the biggest hurdle but once the creator received approval, they were able to set up an account and start publishing code. The open source community understands open source code is not always perfect and it’s up to the community to help improve it over time.