OpenForum Europe

23 January 2023

Feedback to the Commission regarding
Cyber resilience act – new cybersecurity rules for digital products and ancillary services

The CRA and Open Source Software

Dear Sir/Madam,

I am writing on behalf of OpenForum Europe (OFE), a think tank for open source software and open technology policy. This submission is also supported by the Eclipse Foundation, Open Source Initiative (OSI), APELL, CNLL, and The OSB Alliance.

We have very serious concerns regarding some parts of the proposed text in its current form. That said, we appreciate certain aspects and the general goals of the proposal for a Cyber Resilience Act (CRA) and we look forward to working with the EU institutions to strengthen cybersecurity in the EU.

Free and open-source software exemption

We appreciate the intent of Recital 10 to create an exemption for free and open-source software, but we would like to highlight three issues which we feel need to be addressed for this exemption to fulfil its purpose.

This exemption is in line with the Commission's efforts to support the development and use of free and open-source software (European Commission Open Source Software Strategy 2020-2023), however, our first issue is that the exemption is presented as existing "to avoid hampering research and innovation". This mixing of two activities (“free and open-source software” and “research and innovation”) could lead to the exemption being given a very narrow interpretation. And uncertainty about how it will be interpreted makes it difficult for anyone to confidently rely on the exemption. To remove this issue, the wording should begin with "In order not to hamper development or distribution of free and open-source software...".

A second issue is the legal form. To give people, including businesses, enough confidence to rely on this exemption, it must be made into an article, and the "should" must be changed to a "shall".

The third issue is the limitation to acts "outside the course of a commercial activity". With the flexibility of free and open-source software, and with the collaborative approach to its development, there are many different relationships and roles between the various developers and the various publishers or redistributors of a single piece of software. This includes contractors and volunteers and businesses providing support and even individuals who may later receive a bug bounty (such as the European Commission's Open Source Programme Office's bug bounties) or those participating in a sponsorship scheme. If the

definition is too narrow, or if people cannot discern whether they are inside or outside of "the course of a commercial activity", then people will not be able to avail of this exemption and it will fail to fulfil its purpose.

We would be open to discussing how this could be improved. One route we would like to explore is how the exemption could be implemented without relying on the word “commercial”, which has a long history of being difficult to understand in digital and online contexts. We also note that the second half of Recital 10 currently focuses on ensuring that certain users *cannot* make use of this exemption. For the exemption to have an effect, it may be more useful to focus on clarifying who *can* rely on this exemption. For example, by stating that providing technical support services does *not* create a “commercial context” which would affect publication or redistribution of the software.

General incompatibilities with software development

Replacing the current general freedom to publish software with a new system that imposes a set of CRA requirements constitutes a significant disruption to open innovation in Europe. The current formulation of the CRA interferes with almost every software development model other than the case of a single company developing the entire code-base behind closed doors and making periodical releases. This model was common until the late 1990s, but much less so now.

Further, by attaching responsibilities and liabilities (for publishers and redistributors and, when Article 16 applies, for contributors) this proposal makes collaborative software development difficult because one manufacturer will have to assume responsibilities and liabilities for the development performed by their collaborators, over whom they may not have authority to impose or verify adherence to the CRA.

Free and open-source software is a global commons which brings the best minds around the world to collaborate to solve critical challenges. The current CRA requirements do not fully reflect this global nature and societal role and thus the need to protect the current free flow of open code across borders.

Another potential issue is a threat to the resilience of this open code should any software project from outside the EU, if the developers are not interested in the EU market and thus CRA compliance, will not be available to users in the EU. The EU will thus be cut off from the rest of the world and will have access to less technology. This could even include security upgrades, as described below.

Lastly, since publishing software would include the risk of a fine, and would burden the publisher with liabilities and obligations, there will be software which is developed, and which could benefit others, but the author will decide not to publish. Or to not publish in Europe.

One possibility would be to remove the general ban on publishing and instead have CRA compliance as an optional seal of quality. Instead of making the CRA mandatory for publishing software, it would be mandatory for claiming that a particular version is "CRA compliant (CE)". This could be combined with national or EU laws requiring public

administrations, or other specific entities, to only use CRA compliant software. (But before requiring anyone to only use CRA compliant software, the security issues created by the CRA, described below, should be kept in mind.) This approach has the advantage of ensuring that software can still be published, and that the EU will not be cut off from the rest of the world. It also has the benefit of allowing a CRA security audit & certification industry to develop in Europe.

A second approach would be a variant of the first, where the CRA's requirements apply in general but do not apply to free and open-source software because improved security is already enabled by publication of human-readable source code and a licence which permits security audits and publication of modified versions of the software. For this approach, there should of course be a procedure for such available software to obtain CRA & CE certification, if desired.

A third approach would be to place the CRA requirements on someone other than the manufacturers and distributors. By removing this blockage, the EU would no longer be cut off from the rest of the world and developers could continue their work. For this to have real benefit, it must be sufficiently detached from the manufacturer so as to allow a third-party to obtain the software, perform a CRA review, produce the required documentation and make any changes necessary, and then obtain CRA compliance. We would be interested in participating in a dialogue on where to place the requirements—possibly linked to a set of commercial activities or on types of use.

A fourth approach would be to significantly refocus the CRA's scope to specific product categories—coupled with various operational improvements to proposed obligations—so that the regulatory system could first be deployed, improved and, at a later stage, through implementing acts, be expanded to include other product categories. Removal of "components" and "standalone software" would further improve the workability of this proposed regulation.

Blocking the flow of security updates and patches

If a security vulnerability is discovered in software used in Europe, the liability and technical requirements of the CRA, in its current form, would place a hurdle in front of anyone in Europe working on a fix. Furthermore, Europeans will not have access to any security fixes from outside of the EU if those external developers have not followed the CRA's requirements. European governments, companies and citizens thus risk being stuck using vulnerable software, which is surely not the intention of such legislation.

Interfering with security testing

When a vulnerability is discovered, it can be useful to test previous versions of the software for this vulnerability. This can help identify when the problem was created, which can help pinpoint which code changes should be examined to locate the source of the vulnerability. It is also useful to identify which versions contain the vulnerability so that fixed versions can be made available for all vulnerable versions and so that the relevant users can be informed of the need to upgrade. However, the CRA's requirements to stop making vulnerable software

available would make these security procedures impossible in many cases, especially where an open community is performing the triage. Reporting procedure and timing, for example, must be consistent with those established under NIS2 (24 hours / 72 hours) and other related EU cybersecurity legislation would need to be withdrawn to align with CRA's horizontal objectives (e.g. RED delegated act on cybersecurity).

Simplifying the CRA requirements

Another approach to consider would be to simplify the various requirements. However, the required simplification is quite extensive. For projects to be able to operate in a way similar to today's regime of freedom to publish, the CRA requirements would need to be reduced to the point where they can be either (1) fulfilled automatically (through processes and software systems), or (2) fulfilled by the ensuring availability of human-readable source code and a licence as described above, or a combination of (1) and (2).

Uncertain aspects for hosting sites

Code hosting sites, including for-profit companies and self-hosted services belonging to individuals and organisations, face uncertainty in current CRA definitions. They could be understood as "distributors" within the CRA, despite playing a role in software development distinct from app stores and finished-product software. Existing legislation, namely the Digital Services Act and the Copyright Directive, has provided conditional liability exemptions for these sites. The CRA should clarify the conditions for a code hosting site to be considered a mere conduit with minimal or no CRA obligations.

Conclusion

OFE, the co-signatories, and many open source experts and stakeholders are eager to work with the European Commission and the co-legislators to achieve the important goals of the CRA, while maintaining the open source innovation model's benefits, not only for individual developers and companies, but also as a strategic tool for enabling Europe's digital future.

Co-signatories:

- **Eclipse Foundation AISBL**, global open source foundation based in Europe, providing vendor-neutral governance for open source, projects, collaborations, and innovation.
- **Open Source Initiative** (OSI), global stewards of the Open Source Definition (OSD)
- **CNLL**—le Conseil National du Logiciel Libre / the union of open digital businesses in France
- **The OSB Alliance** - Bundesverband für digitale Souveränität e.V. (Federal Association for Digital Sovereignty) represents around 200 member companies of the open source economy, which together generate more than 1.7 billion euros annually in Germany.
- **APELL** (Association Professionnelle Européenne du Logiciel Libre) is Europe's Open Source Business Association. Founded in 2020 to bring national Open Source Software ('OSS') organisations together into a European network to provide them

with peer support and collective marketing, as well as capacity building and policy support for public affairs, both nationally and on the EU-level.

Yours sincerely,

Ciarán O'Riordan,
Senior Policy Advisor, OpenForum Europe
