



Task Force 2  
Our Common Digital Future: Affordable, Accessible  
and Inclusive Digital Public Infrastructure



INDIA 2023



भारत 2023 INDIA

# RESILIENT DIGITAL INFRASTRUCTURE: ADDRESSING SOFTWARE SUPPLY CHAIN VULNERABILITIES

June 2023


Divyansha Sehgal, Researcher and YLT Fellow, Centre for Internet and Society

वसुधैव कुटुम्बकम्

ONE EARTH • ONE FAMILY • ONE FUTURE



# **Abstract**



**F**ree and open-source software (FOSS) components are the bedrock on which our digital infrastructure is built.

Most software—be it code that logs a user into their phone in the morning or checks the weather, or government systems that authenticate digital identity, streamline payments, and distribute health benefits—use public code written by volunteer developers as part of their codebases. Code reuse is a common practice in software development where large software projects are made up of a collection of public projects so that developers and companies do not reinvent the wheel every time they need to perform ancillary tasks. Despite the well-known practical benefits of code reuse and its prevalence in all digital products and services, several security incidents in widely used FOSS projects have shown that such projects are often underfunded and poorly maintained.

Such lapses are opportunities for targeted interventions in both technical and social aspects of OSS security. Policy solutions can help treat FOSS as the digital infrastructure that it is, by investing in maintaining critical software components used by the government and industry.


For software being created for government and public service initiatives like digital identity or welfare distribution platforms, efforts can be made to compel vendors to contribute to the maintenance of FOSS components they use, further strengthening the ecosystem they draw from. With the governments participating in and supporting the existing open-source communities, they can contribute to sustaining and nourishing an existing pool of expertise that is already passionate about the security and resilience of the software they create.<sup>a</sup>

---

a The author thanks Isha Suri, Divyank Katira and Upasana Hembram for their comments and suggestions on this brief.



# The Challenge



# 1




## Background: What is FOSS?

Code reuse is a common practice in software development, where large software projects rely heavily on existing public projects to implement their composite functionality. Most software—be it code that logs a user into their phone in the morning, reports the day’s weather, or government systems that authenticate digital identity, streamline payments, and distribute health benefits—all use public code written by volunteers as part of their codebases. This public code is known as free and open-source software (FOSS) or sometimes free/libre/ open-source software, and refers to software that is freely available to anyone for consumption, modification, and redistribution. (There are ideological differences in the use and meaning of each of the terms ‘free’, ‘open’, and ‘libre’. For the purposes of this paper, we use the definition provided by Open-Source Initiative<sup>1</sup>)

It is this free sharing of code that makes most modern digital products work. Once a developer has solved a particular problem, such as encrypting data for secure transfer over the internet, they package the solution and

make it available for other developers to use in other products under a permissible license. This enables other developers to use this package to encrypt their data without having to reinvent the wheel each time. This free and open sharing of coding languages, software libraries, components, and projects enables the formation of companies that can use this existing digital infrastructure to build products and provide services to customers.

While the exact economic impact of FOSS is hard to measure because products are not required to disclose their FOSS usage, it is obvious that this digital infrastructure is both advantageous and necessary. Some industry estimates suggest 96 percent of codebases use open-source software as a building block.<sup>2,3</sup> GitHub, one of the most popular platforms where open-source code is hosted, recorded over 413 million contributions to open-source software in 2022.<sup>4</sup> Reports also show time and again that industry leaders and technical stakeholders prefer to increase the adoption of FOSS in their enterprise products because of lowered costs and perceived security benefits.<sup>5</sup> A report released by the European Commission in 2021 estimates the contribution



of FOSS to the Eurozone GDP to be between €65 and €95 million.<sup>6</sup>

FOSS developers recognise the significance of their contributions. The main motivation to contribute to open-source software is not monetary, even in cases where developers are being paid by their companies to develop the software. Instead, developers like to be in a community with other like-minded people to solve challenging problems, build reputation, and learn new skills. Developers also believe in the mission of ‘free’ and ‘open’ software and have a deep commitment to their projects.<sup>7</sup>

### The FOSS security challenge

Open-source software is assumed to be more secure since, theoretically, there are more eyes on the code and an active community which will catch defects better than any individual could. But, this may not always be the case. Several high-profile security incidents in widely used open-source software projects have drawn attention to the fact that such projects are often underfunded and poorly maintained, meaning that maintainers are unable to catch security vulnerabilities or update components in a timely manner.

This can have severe consequences because the widespread use of open-source software means that a vulnerability in a software component or a library can have cascading impacts on all the tools that use it.

For example, in 2014 it was discovered that a popular open-source encryption library which is used by governments and technology companies around the world had a programming error that allowed hackers to intercept confidential data. The bug, called Heartbleed, affected almost half a million websites online given that it was a trusted, widely used software library.<sup>8</sup> There has also recently been an uptick in vulnerabilities discovered in the software supply chain. An example is the Log4Shell attack in 2021 which allowed attackers to execute arbitrary code on vulnerable servers and affected millions of devices worldwide.<sup>9</sup>

The media attention to the Heartbleed bug also revealed that the critical library was maintained by a small team of volunteer coders with only one full-time developer and minimal financial resources to support their work, which left the library open to vulnerabilities.<sup>10</sup>

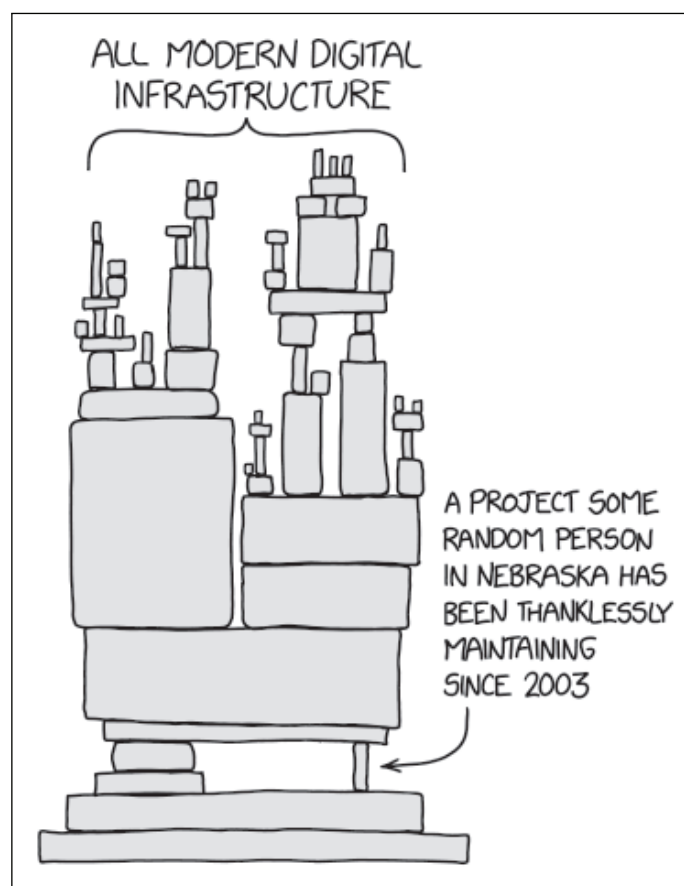
The digital security challenge that the software supply chain faces is

not unrelated to the infrastructural challenges and incentives that drive contribution to and adoption of FOSS. Most open-source software is created and maintained by individuals and/or small teams. While almost all software contains open-source software, reports have shown that 94 percent of OSS projects have fewer than 10 developers writing code.<sup>11</sup> This makes 'Figure 1'

particularly popular in FOSS circles as a succinct description of the problem.

Another complication is that FOSS varies widely in quality, and open-source software skills are not just 'one skill' or 'one coding language'. While individuals can contribute to multiple projects, funding one does not mean that other FOSS projects gain any benefits from that funding.

**Figure 1: Dependency**



Source: *xkcd*<sup>12</sup>



Research has found that FOSS projects are largely unreviewed for security issues, lack adequate systemic safeguards to prevent tampering of code, and do not present users with the tools to verify whether the software they are consuming matches the expected source code.<sup>13</sup> Further, there is no standard way to monitor FOSS components and update them when a vulnerability is discovered. This means that once a vulnerability is identified in a particular library, software developers and security teams will often need to scan through entire codebases and dependencies to identify and isolate each individual use of a library to be able to update it.

Among technical stakeholders who create software applications, security of the code is often seen as a secondary priority to the expected functionality

in day-to-day operations. If a software component has the functionality required, security testing is of lower priority.<sup>14</sup> Further, the use of OSS is often promoted as a cost-cutting exercise and is painted with a consumer attitude which prioritises the ability to quickly resolve problems among companies, and even government policies.<sup>15</sup>

There needs to be a recognition that the use of FOSS does not mean that companies and governments get to cut costs in their digital projects. Instead with the use of FOSS, they become part of the broader community that benefits from the continued existence and timely updates of the project chosen. Thus, the key to strengthening security is to strengthen the ecosystem of users and contributors who interact with open-source software.



# The G20's Role

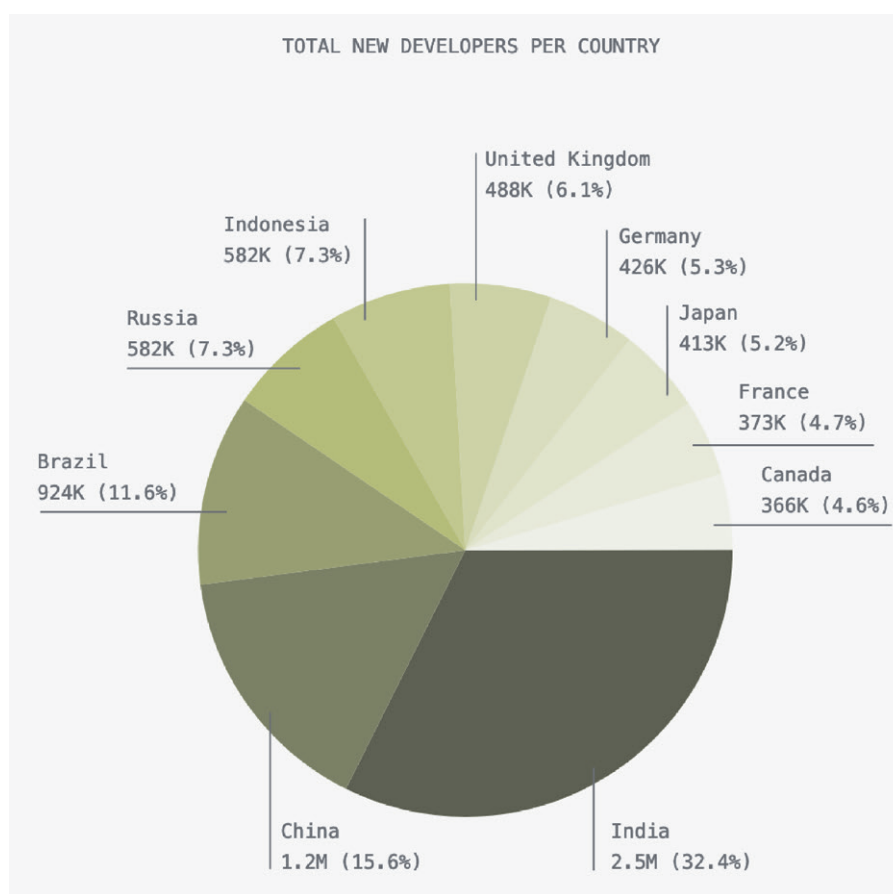
# 2

The development and use of FOSS is not restricted by geographical boundaries, making the G20 the ideal forum for taking action on securing the software supply chain. Developing and contributing to open source is a global endeavour. GitHub attracts developers from all over the world, with India showing the largest year-over-year increase (32.4 percent) in developer populations on the platform.<sup>16</sup> Other G20 countries such as Brazil and China

are also extremely well represented on the platform (Figure 2).<sup>17</sup>

While not a perfect metric, it can be used to infer open-source software's global popularity. Since FOSS developers are geographically distributed, problems related to FOSS security also warrant a global response. Global investment in open-source software and its security presents an opportunity to nurture intentional international collaboration.

**Figure 2: The state of open-source software**




Source: Github<sup>18</sup>



# **Recommendations to the G20**



# **3**



**S**upply chain security issues with open-source software (e.g., undetected vulnerabilities in widely used libraries) are also the problems facing open source in general (lack of resources and inadequate maintenance). Therefore, to secure the software supply chain for our digital infrastructure, interventions need to be made on two main levels:

1. Proactively securing software projects that are critical for everyday operations
2. Developing and sustaining a robust open-source culture and support infrastructure

While the recommendations listed are not comprehensive, they are a starting point towards a more secure software supply chain.


### **Proactively securing existing software projects**

1. **Identify the most commonly used projects in government and industry.** There is an urgent need to identify projects and libraries that are commonly used and where they are located in the software

systems of each company and product.<sup>19,20</sup> While GitHub does have some popularity metrics, they are insufficient to identify which FOSS components are in the highest demand. A public-private partnership (or an industry collaboration, e.g., OpenSFF<sup>21</sup>) is necessary to identify which FOSS projects are the most used across digital products. This must be an internationally collaborative endeavour because the creation and implementation of code is not bound by the geography of the developers.

2. **Invest in securing and maintaining these projects.**

Creating a list of these commonly used projects would help governments invest in, support, and co-create projects they use most frequently. It would also help projects raise money from external funders and deploy it towards maintenance and server costs and recruit team members to critical projects. Support need not only be financial: providing training resources, upskilling on security best practices, helping projects plan succession workflows once team members move on, and stepping up design and marketing



are all ways that can support a small team of FOSS maintainers. There are currently some civil society initiatives like FOSS United<sup>22</sup> and industry initiatives like GitHub Sponsors<sup>23</sup> that aim to fund FOSS projects, however this funding is currently piecemeal and ad hoc.

3. **Periodic ongoing third-party audits of critical projects** are important to identify security issues in the most used FOSS projects so that they can be fixed as common international infrastructure. Some companies perform this function for their clients, while non-profit and industry collaborations like OpenSSF and the Linux Foundation projects such as Alpha-Omega<sup>24</sup> are also starting to investigate the larger industry and compile software library level data. An international public-private partnership that periodically audits critical software for vulnerabilities is necessary.

4. **Communicate identified software vulnerabilities and their fixes.** Similar to identifying critical projects, a multistakeholder partnership that keeps track of known software vulnerabilities and communicates fixes to users is

necessary.<sup>25</sup> There are government, industry, and non-profit bodies that keep track of known software vulnerabilities when they are submitted by volunteers, e.g., the Common Vulnerabilities and Exposures database.<sup>26</sup> National Computer Emergency Response Teams can play a huge role in the communication of software vulnerabilities and their fixes identified through ongoing audits.

5. **Incentivise the creation and use of a software bill of materials (SBOM) for all public sector projects** from vendors. An SBOM keeps track of all dependencies in a digital product so that once a vulnerability is identified in a FOSS component, it can be located and fixed by upstream users more quickly. Making these a matter of public record would also enable the community to identify and communicate security issues, would help developers streamline the implementation of fixes, and get affected systems back online quickly.<sup>27</sup>

6. **Ensure security education for software developers.** There is an urgent need to educate developers of FOSS and enterprise software on security best practices like the



use of memory safe languages for development, fuzz testing projects, securing releases cryptographically, validating existing dependencies, and investing in FOSS management tools to ensure secure distribution and loading of FOSS components.<sup>28</sup> Research has shown that for developers, delivering a functional product is often the first priority.<sup>29</sup> Given the high-stakes associated with non-secure software, security education can help ensure that software security does not remain a secondary priority.

## Developing and sustaining an open-source culture

Developing and sustaining an open-source culture is important in ensuring the security of FOSS. A review of the literature suggests that there is space for targeted interventions in the following categories, which could have cascading effects on the health of the open-source community, and consequently the digital public goods that are created through its efforts.<sup>30,31,32,33</sup>


### 1. Government OSS consumption:

The government as a large consumer has market-moving

and trendsetting potential for FOSS communities.

- Digital policies should focus not only on code usage but also on contribution to the projects used. This can be done by prioritising vendors that contribute to existing projects and co-creating software products with the open-source community directly.
- Procurement policies should mandate that software used for public services should be open-source, not proprietary. If open-source versions are not available, vendors should aim to release infrastructure products under an open-source license so that the entire community can benefit.
- Establish an Open-Source Program Office that coordinates the implementation of open-source software across government departments. This office can be a knowledge centre within the G20 states which can advise on OSS best practices, provide training, and aid adoption.

### 2. FOSS contributions from the technology industry:



Incentivising FOSS contributions from technology companies too has the potential to create high-impact, high-quality open-source projects with a dedicated community around them. For example, ‘React’<sup>34</sup>, a web library that helps build user interfaces online, is maintained by Meta and is one of the most popular front-end frameworks.

- 3. Education:** Curriculums in high schools, colleges, and postgraduate institutions should adopt a FOSS-first approach

to build FOSS skills in the next generation of software engineers. Education can play a major role in FOSS adoption, by inculcating FOSS values among the next generation of technology creators and keeping developers up to date on the latest security thinking and best practices. Creating FOSS communities in institutes of higher education is a great way for students to be more involved with the larger FOSS community, contribute to existing projects, and see the impact of their contributions.

Attribution: Divyansha Sehgal, “Resilient Digital Infrastructure: Addressing Software Supply Chain Vulnerabilities,” *T20 Policy Brief*, June 2023.

## Endnotes

---

- 1 “The Open Source Initiative,” Open Source Initiative, *Accessed April 4, 2023*. <https://opensource.org/osd/>
- 2 “2022 Open Source Security and Risk Analysis Report,” Synopsis, 2022. <https://www.synopsys.com/content/dam/synopsys/sig-assets/reports/rep-ossra-2022.pdf>.
- 3 “2023 Open Source Security and Analysis Report,” Synopsis, 2023. <https://www.synopsys.com/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html>.
- 4 “Octoverse 2022: The state of the open source software,” Github, *accessed Jun 15, 2023*. <https://octoverse.github.com>
- 5 “The State of Enterprise Open Source: A Red Hat Report,” Red Hat, 2022. <https://www.redhat.com/en/resources/state-of-enterprise-open-source-report-2022>.
- 6 Knut Blind, Sivan Pätsch, Sachico Muto, Mirko Böhm, Torben Schubert, Paula Grzegorzewska, Andrew Katz. 2021. “The impact of open source software and hardware on technological independence, competitiveness and innovation in the EU economy – Final study report,” Publications Office, European Commission, Directorate-General for Communications Networks, Content and Technology. <https://data.europa.eu/doi/10.2759/430161>
- 7 “Report on the 2020 FOSS Contributor Survey,” Linux Foundation, 2022. <https://www.linuxfoundation.org/resources/publications/foss-contributor-2020>
- 8 Joseph Steinberg, “Massive Internet Security Vulnerability – Here’s What You Need To Do.” Forbes, 2014. <https://www.forbes.com/sites/josephsteinberg/2014/04/10/massive-internet-security-vulnerability-you-are-at-risk-what-you-need-to-do/?sh=3207221c3fdf>
- 9 David Uberti, James Rundle, Catherine Stupp, “The Log4j Vulnerability: Millions of Attempts Made Per Hour to Exploit Software Flaw,” The Wall Street Journal, 2021. <https://www.wsj.com/articles/what-is-the-log4j-vulnerability-11639446180>
- 10 Nadia Eghbal, “The Unseen Labor Behind Our Digital Infrastructure.” Ford Foundation, 2016. <https://www.fordfoundation.org/work/learning/research-reports/roads-and-bridges-the-unseen-labor-behind-our-digital-infrastructure/>
- 11 Frank Nagle, James Dana, Jennifer Hoffman, Steven Randazzo, Yanuo Zhou, “Census II of Free and Open Source Software – Application Libraries” The Linux Foundation and The Laboratory for Innovation Science at Harvard, 2022. <https://www.linuxfoundation.org/research/census-ii-of-free-and-open-source-software-application-libraries>



- 12 “Dependency”, xkcd, accessed June 15, 2023, <https://xkcd.com/2347/>
- 13 Divyank Katira, “Securing Our Dependence on Code Reuse in Software.” Centre for Internet and Society, April 2023. <https://cis-india.org/openness/securing-our-dependence-on-code-reuse-in-software>
- 14 Divyansha Sehgal, “Security of Open Source Software: A Survey of Technical Stakeholders’ Perceptions and Actions,” Centre for Internet and Society, April 2023. <https://cis-india.org/openness/security-of-open-source-software-a-survey-of-technical-stakeholders2019-perceptions-and-actions-1>
- 15 Apar Gupta, “Analysis of FOSS Government Policies in India,” 2022. <http://dx.doi.org/10.2139/ssrn.4146240>
- 16 Github, “Octoverse 2022: The state of the open source software.”
- 17 Github, “Octoverse 2022: The state of the open source software.”
- 18 Github, “Octoverse 2022: The state of the open source software.”
- 19 Divyank Katira, “Securing Our Dependence on Code Reuse in Software.”
- 20 “The Open Source Software Security Mobilization Plan.” Whitepaper. OpenSSF, and The Linux Foundation, 2022. <https://openssf.org/oss-security-mobilization-plan/>.
- 21 “Open Source Security Foundation (OpenSSF),” OpenSSF, accessed March 24, 2023. <https://openssf.org/>.
- 22 “Funding and Grants,” FOSS United, accessed Jun 15, 2023. <https://fossunited.org/grants>.
- 23 “Github Sponsors : Invest in software that powers your world,” Github, accessed Jun 15, 2023, <https://github.com/sponsors>.
- 24 “Alpha-Omega,” OpenSSF, accessed Jun 15, 2023. <https://openssf.org/community/alpha-omega/>
- 25 OpenSSF and The Linux Foundation, “The Open Source Software Security Mobilization Plan.”
- 26 “Overview,” Common Vulnerabilities and Exposures, accessed Jun 15, 2023. <https://www.cve.org/About/Overview>
- 27 Amelie Koran, Wendy Nather, Stewart Scott, and Sara Ann Brackett. “The Cases for Using the SBOMs We Build.” The Atlantic Council, 2022. <https://www.atlanticcouncil.org/in-depth-research-reports/issue-brief/the-cases-for-using-sboms/>.

- 28 Divyank Katira, "Securing Our Dependence on Code Reuse in Software."
- 29 Divyansha Sehgal, "Security of Open Source Software: A Survey of Technical Stakeholders' Perceptions and Actions."
- 30 "The State of Free and Open Source Software in India," CivicDataLab, 2022. <https://state-of-foss.in/the-state-of-foss-report.pdf>.
- 31 Nadia Eghbal, "The Unseen Labor Behind Our Digital Infrastructure."
- 32 Apar Gupta Arjun Gargeyas, Bharat Reddy, Kailash Nadh, Nitin Pai, Pranay Kotasthane, Rushabh Mehta, Saurabh Chandra, and Venkatesh Hariharan. 2023. "An Open Tech Strategy for India. Version 1.2." Takshashila Institution. <https://takshashila.org.in/research/an-open-tech-strategy-for-india>
- 33 Stewart Scott, Sara Ann Brackett, Trey Herr, and Maia Hamin, "Avoiding the Success Trap: Toward Policy for Open-Source Software as Infrastructure," The Atlantic Council, 2023. <https://www.atlanticcouncil.org/in-depth-research-reports/report/open-source-software-as-infrastructure/>.
- 34 "React," React, Meta Open Source, *accessed March 24, 2023*. <https://react.dev/>





वसुधैव कुटुम्बकम्

ONE EARTH • ONE FAMILY • ONE FUTURE