SUBCOMMITTEE ON INVESTIGATIONS AND OVERSIGHT AND SUBCOMMITTEE ON RESEARCH AND TECHNOLOGY
SECURING THE DIGITAL COMMONS: OPEN-SOURCE SOFTWARE CYBERSECURITY

Ms. Amélie Erin Koran
May 11, 2022

Chairman Foster and Chairwoman Stevens and members of the Subcommittee, thank you for the opportunity to testify before you today. I am Amélie Koran, Non-Resident Senior Fellow in the Cyber Statecraft Initiative at the Scowcroft Center for Strategy and Security at The Atlantic Council. It is an honor and a pleasure to be here with Dr. Lohn and Mr. Behlendorf.

At the Cyber Statecraft Initiative, we work at the nexus of geopolitics, technology, and security to help shape policy and better inform and secure the users of technology. This work takes place in three clusters, the Geopolitics of Cybersecurity, Securing Operational Technology, and Communities of Cyberspace. The Initiative strives to address strategic questions by combining systems analysis, policymaker engagement, and the operational experience of our interdisciplinary practitioner community.

In my opening remarks I'd like to discuss the impact of realistic and applicable actions that can be used to address better securing the open-source software ecosystem and how to educate developers and users more effectively and responsibly. My views and perspectives come from a point of a contributor to open-source projects, a technician who's worked in securing and operating systems in critical infrastructure, and one lucky enough to experience all of this within both public and private sectors at various levels and in different industries.

Code is Speech and Infrastructure

The concepts of open-source software were not only intended to be something to free developers and creators of software from the shackles of onerous licensing terms common for contemporary computing, but also a way to allow them, in a way that was natural for them, to freely express speech through code. The counterculture of the 1960s, mainly hobbyists and research scientists, homed on university systems and networks supported by academia and the government, felt that the best way for technology to advance was sharing. While not on these networks, they met up at "fests" to share their hacks to get new capabilities out of systems they had available, or novels ways to improve what they had access to.

Fast forward a few decades, when personal computing started to take hold, more of this free-ware made it into the hands of consumers, it was often provided with an ask that if those users found it useful, to possibly drop a contribution, in the way of monetary remuneration or fixes to bugs, and thus a "share-a-like" model was born, and the alternative to copyright, a copyleft, license model came into existence. These requirements of these licenses prioritized sharing contributions, embracing the logic that many eyes looking and working on such software would more rapidly surface bugs and fixes, so long as these efforts came back to the core code.

As access to the Internet expanded, and such code became widely accessible, more and more projects, from applications to the core operating systems on which they ran, were available under these very permissible licenses known often as open source. Most, if not all, were free with that caveat of a responsible user will contribute back to the code in line with the selected license of the original author. Some of these projects, as they grew, began to adopt governance models to support larger projects through management of resources, commits to the core code base, as well as laying out road maps for features and other changes.

This core concept of project governance is one of the greatest challenges, yet also greatest opportunities in which this committee and the government can assist the open-source movement. Governance is both a rudder and engine for projects, providing direction and velocity, via an agreed upon route and stop or gates along the way to make sure everything is coming along as planned. JFK setting the goal of the US space program to land a man on the moon before the decade of the 1960s was out, was a form of governance. He gave the goal, set where the US technology efforts were to be focused, and worked with his partners in government to resource the activity to achieve the goal, even if that leadership mantle was passed on. The process of how government evaluated progress and how that aligned with overall policy of the US also composed governance in that instance. It doesn't always have to be overbearing but can also be inspirational. We need just this manner of governance now.

One of the original progenitors of the concern both in how much our modern society relies upon open-source code, but also provided the most stereotypical example of some of the most common failings was the Heartbleed vulnerability disclose din early 2014. I had the opportunity, through what some may consider luck, but also circumstance, to observe and participate in our government's approach to handling this incident while on a leadership development rotation at the Office of Management and Budget.

One of the challenges which hampered a more comprehensive approach to triaging and responding to this event by the US Government was not understanding the vulnerability in a comprehensive manner. This was due to those in charge, having been only aware to the surface use of such technologies. The vulnerability ran much deeper than initially expected, but the lack of experience and actual technology literacy of the response coordinators and policymakers wasted time and resources in the initial response.

While websites were a major part of our core digital economy and often the most visible public face of the internet, the vulnerability in OpenSSL impacted the internet's core infrastructure – its underbelly; compiled in the operating systems of the routers and switches, in the protocols, which made that lock icon useful. In short, we had a "baked in" issue with near billions of devices, as well as applications, which made response and resolution more challenging that merely patching an offending application. This challenge of confronting cracks in the foundation of digital infrastructure would return, most recently in the Log4j response.

What made Heartbleed so challenging, versus simply picking up the phone or dropping an email to a vendor such as a Microsoft, Google, Cisco, Amazon or Apple to ask them to modify their proprietary, non-open-source code, was that this was a project largely maintained by volunteers working in their free time out of interest. personal values, or the utility this code had for them. That personal value and utility statement carries true for the creation of a lot of open-source software we are familiar with today.

Open-Source Values and Governance

While we are gathered here to discuss out how to support the open-source community and foster this ecosystem, the phrase "we're from the government, and we're here to help" is somewhat an inhibitor. Government should foster collaboration and create venues and opportunities for that, not creating another checklist or reporting mandate that adds more work or confuses the desired outcomes of securing critical open-source software.

The executive order from May of last year was a way to have agencies to conceptualize their challenges with managing open-source use and application of such technology in their environments but does very little to assist or address it anywhere else. It is a dark cloud over agencies and may stifle innovation and self-determination, but also puts a chill over industry as it was so focused on Federal entities without expressing stronger needs to collaborate with the open-source community and supporting facilities.

We're here to discuss the best way for the agencies and their associated missions and programs can best support this challenge. This is not just an all of government problem to solve or address but is international. Few in this space have actively stepped up to take the reins. The world has witnessed our digital interdependency throughout the war in Ukraine, efforts to secure systems there have made Americans and our allies safer. In open-source software, there is another opportunity for the United States to be a global leader and obtain some of the American exceptionalism back in the global community as well as the open-source ecosystem. Potential actions by Congress as well as agencies fall in line with regulation, standards creation, sector coordination, and even grantmaking efforts. Our challenge exists in figuring out where they can best interface and, quite literally, get the best bang for the buck.

As I had my time at agencies, and most notably at Health and Human Services, which holds the title of the largest grant-making authority in the world, it also contains the largest inspector general for oversight in all of the US Government, to ferret out waste, fraud and abuse. Add to the perception that the cybersecurity industry has become filled with false or overleveraged promises, guaranteeing to chase after every event and incident touting their wares. Adding a pot of money in the wrong hands or wrong place may attract many more bad actors to what is essentially a gold rush exacerbated by recent incidents from Solarwinds to Log4j, with many more destined to come.

However, this provides a good opportunity to engage private sector and non-profit entities already established to help interface with open-source software projects on a level where these resources can be guided after professional evaluation and management into the right hands where they can do the most good.

We are joined by one such organization, the Open Secure Software Foundation, under the wing of the Linux Foundation, a not for profit established to help manage, maintain, and govern several key open-source, critical software projects. Just a few months ago, the Apache Foundation, which maintains several other essential software projects, and truly is an excellent example of longstanding and scalable governance frameworks joined the Senate to discuss open-source software and Log4j. But these organizations are rare when you look at the entire open-source ecosystem. Self-interest from a foundation or other similar organization may occur, however subtly, by prioritizing suggested changes, features, or even direction by those who provide resources such as funding or staff time, at the detriment of addressing or solving something in a more democratic or egalitarian way from a less potentially partisan leadership.

Very few projects and code bases reach the scale to where they are lucky enough to become funded, managed and governed by foundations like these. Some may be given resources by consumers of their code, potentially from larger organizations that benefit from not having to pay licensing but feel it's in their best interest to share back to help keep projects healthy, but often nothing formalized as to who and how features, modifications, and versions are planned and delivered. These are generally the ninety-nine percent of open-source software projects, regardless of their perceived usefulness or critically to the proper operation of our digital economy and infrastructure.

That one percent, curated by foundations and other support models, often, though maybe not as transparent, are often beholden to the whims and wishes of their board benefactors, which come in the shape, in most cases from large technology companies that have integrated their code into their own products and services, thus creating a self-interest which is in opposition of organic and self-sustaining nature of open-source software. In short, this takes many parallels to the old fire companies of large cities prior to the American Civil War, where response was prioritized for those who paid your fire companies and was not a public good provided by the government.

This is something our discussion here should begin to address, which is to help find ways to triage critical, core, open-source digital infrastructure and provide the guidance necessary to engender trust in the use and utilization of it, but ensure that it's care and feeding is addressed as the public good it was intended to be, rather than be beholden to what resources are applied to it by foundation grants at the commercial level. We do have to tread carefully, as this may result in locking up future investments by those private sector technology organizations, so an opportunity to coordinate and align should be a first step.

Standards and Validation

While we look to NIST and the NCCoE as an essential player in interfacing with the open-source community in the services it provides best, which is guidance and standards, we also need to lean on their methodology for assessments and validation, such as in use for the FIPS encryption process. The Special Publication series, colloquially known as the "SPs", have been some of the most effective national and international contributions to computer security the US government has created, and industry has voluntarily adopted or referenced. In my time as both a public servant, but also private sector employee, nearly every company and organization has used various SPs to use as a bar to reach or be measured by for compliance and addressing of gaps in their configurations and operations of technology environments.

This is often due to organizations' desires to work with government, and the requirements in many cases that systems be compliant to these standards and guidance, but also, in lieu of comprehensive best practices developed by industry, since many technology environments are hybrids from many vendors, it is the only holistic method to utilize. However, this bar that is reached has been addressed as the high bar, rather than the minimum base to secure or mitigate threats to systems. This leaves many without the resiliency to take the eventual hit from a breach, attack, or other adverse event.

While the SP series addresses aspects of these systems and technologies in use, gaps remain for where this can assist making open-source software more secure. Noted earlier, governance is a major component to success and long-term viability of open-source projects. What can be proposed here is to develop guidance that can be adopted by projects, large and small, like a "what to expect when you're expecting an open-source project" book like you have for expectant parents, that provides tools and guidance on how to structure, build, operate and maintain such activities. As NIST does, to convene experts to contribute to this guidance. It would be a good first step to be able to offer the open-source software community at least a framework which projects at various stages can look to achieve or conform to, in this case a standard or guide for open-source project governance.

Leveraging the well-worn process for validation, and the deep reach into the private sector for such services, as well as their stewardship of the national vulnerability database, NIST is in an enviable position to share that knowledge and interface with solutions and services already trusted and used by a good portion of the developer and user community for open source. Offering frameworks to prepare key software packages, potentially hosted at locations such as GitHub and GitLab, among others, to go through a vulnerability validation process, or, even as low-level as to provide or support build and test services for critical code bases is a workable way forward for NIST to have an effective role in this space. Providing automated tools and services, those which make sense to automate, checking for well-known or obvious issues, but may not be a capability available to all developers, can free those developers to work on tougher, less-obvious issues that they can address. Services offered to Federal agencies, such as CARWASH for mobile applications, is one example that similarly can be developed and deployed. GitHub recently added and expanded availability of just such tools to committers who utilize their services, including alerting users to insecure dependencies that have been imported into their code bases, a previously resource intensive, manual activity for developers to perform on their own.

Creation of an independent Underwriters Laboratory (UL)-like for critical open-source software programs, similar to what we have for more physical systems, is one path. This is something

Germany has already undertaken as part of their involvement with vulnerability treatment efforts from OECD (Organization for Economic Co-operation and Development), via regional TÜVs (Technischer Überwachungsverein), technical inspection associations, but we have yet to do at scale for software system within the United States. Assurance is the name of the game when wondering if the latest bit of code they opted to utilize will adversely affect the operations of their organization.

This verification lab service like UL, would be voluntary for projects who wish to be used by critical infrastructure, but once through the process, can carry the trusted verification. The process should be agnostic, whether the code is maintained by a non-industry or sector affiliated individual or team, or a large corporation who's chose to create and steward a open source project. Much like you cannot pick and choose which physical infrastructure you should repair based on who it serves, the same model needs to apply here. The NSF in conjunction with NIST are best candidates to develop this process and identify the metrics and measures required. If this gets to a state of international collaboration, this US Government agency partnership merely shall be subsumed as supporting affiliate members within the international community.

Additionally, OSS projects should be providing an easy to use, understand, and apply software bill of materials (SBOM), to assist with decision support for organizations who opt to be open-source software friendly consumers to determine if they picked a healthy solution to base their operations on. SBOMs offer a point in time view for checking the "ingredients list" through advanced software composition analysis, offering up a role for NIST for maintaining a historical record or database of performance over time, that is searchable, similar to the National Vulnerability Database (NVD) which is relied upon heavily for checking the status of known individual vulnerabilities in both open source and commercial solutions, but rarely is used to help analyze and risk score systems that may be composed of multiple packages and code bases, and leave consumers to make best guesses rather than data-based decisions on their consumption of open-source software.

Assessment, Categorization, and Triage

Beyond the highlighted capabilities of the government to convene, collaborate and align resources at a national and international level, it also can muster these resources at scale like no other entity, to support a public need. As seen from disaster response to military power, the typically maligned bureaucracy can be put aside in many cases to quite literally move mountains.

Focusing this ability on a realm the US government is not necessarily the top of the heap in, requires a direct, focused, and patient touch. DHS, in their roles in coordinating sector security such as power, transportation and others, has a unique role in applying guidance, but also working with such sectors to listen and work to correlate and prioritize common needs. For critical open-source software, CISA, NIST and research from NSF programs, should agnostically assess, categorize, and triage the top projects of interest and work with those sector coordinating councils, developers, integrators, and consumers to remediate issues, develop resourcing strategies, and help with project governance. A task force from these agencies and components should be formed to operationalize these first steps until transitioned to a more authoritative office or agency component. This cannot wait for typical legislative processes to hem and haw while these problems grow and are exacerbated daily.

For example, with local telephone companies, or even the US Postal service, for projects that may lack all the above, either due to size, resources, abandonment, or other complication, CISA and its partners essentially may become “carrier of last resort”. They should for a time, help with these efforts and look to match the project in its state at the time with willing supporters to shepherd it to a point where trust, reliability, and resilience can be achieved. Such a government led effort could be well complemented by an established volunteer network of open-source developers and security practitioners, with the existing goal of ‘swarming’ to important but underserved code to mitigate risk.

This carrier of last resort status is literally the “Hail Mary” for identified critical projects or code that have become critical but have lost all means of maintenance and support to keep the projects viable or interest other parties to maintain and continue developing. Sometimes this may be due to the presence of very old code, change in status of a maintainer, or overall lack of interest beyond a release. Any effort to directly interface by the US Government must consider these cases and plan accordingly.

Education and Stewardship

Finally, it is very easy to focus on projects, their developers and the technical nits involved in open-source software security. However, much like many of us should have learned in programs such as home economics in school, being a smart consumer is also paramount into driving adoption and use of such tools in our lives and communities.

As a former Chief Technology Officer, along with Deputy Chief Information Officer and Enterprise Security Architect, I’ve had to consider the ramifications and impacts to systems I was responsible for when selecting a technology strategy for my organization. It’s very easy for many organizations to strictly focus on cost or features but miss the bigger picture of the total cost of ownership which includes looking at the lifecycle of such adoption. This results in many gaps for resources such as maintenance and operations, but also inclusion of knowledge management, training, and awareness for both technology staff, but users who will be interacting with it.

For developers, it’s also not just writing code, but considerations far outside code quality and completeness, and should also dive into the realms of providing methods for interacting with data security and management, privacy, and user experience, which are, albeit abstract, but still components of designing, building, and operating secure systems. Operations and security staff are often already overtasked and under resourced in many organizations, so focusing on the design and build of secure code, whether it be proprietary or open source, helps remove any extra load on that staff, which translates up the chain to leaders and customers of organizations who chose to utilize open source-based solutions.

While foundations can help cover parts of these tasks through selective engagement, guides, frameworks, and badge programs, there are still gaps that need to be collaboratively addressed in partnership between the public and private sector. NIST through programs supported by NICE, provide well-established educational frameworks and interfaces with institutions without having to rework the proverbial wheel to establish relationships and a curriculum. Having the private sector focus foundational resources to work directly with NICE can shorten the time and increase resources it would take to put efforts like those from OpenSSF into action.

For example, by leveraging NICE, versus going alone, programs from OpenSSF and others can focus on developing lesson plans and content, as well as supporting or operating "hack-a-thons" to get ahead of open-source projects that may need a swarm of resources to shore up their security. It will remove the extra labor and time desired by such independent efforts to initially create those connections with our education infrastructure. It is merely one match of many that the Federal government can make to help address these challenges.

Conclusion

While all of this appears at first blush to appear a never ending and daunting task, parts of the solution are in motion, but not entirely aligned or moving at the same speed and rhythm. Noting that government's strengths exist in the power of collaboration and coordination, but also the trust and faith many put into the institution, it just takes the wherewithal and dedication openly, to commit to put the full weight of government and its resources behind it to make it happen. It has now arrived at a point where we can no longer hem and haw about what to do, because technology won't wait or slow down to work at the pace of government, but government needs to act at the pace of technology and iterate its collaborations at its speed to achieve results.

Trust your experts, listen, learn, and build these relationships to help support forward leaning decisions rather than to strictly react. Use the power of automation to help address some of the easy problems, and allow, often the inelastic and unscalable resources, of smart people, try to crack some of the bigger nuts and problems by getting them time to work together and offering venues opportunities to solve by sponsoring such collaboration efforts. Open-source software survives by many people working together to apply themselves to a problem, find a way to work within those models.