SECURING THE DIGITAL COMMONS: OPEN-SOURCE SOFTWARE CYBERSECURITY

JOINT HEARING

BEFORE THE

SUBCOMMITTEE ON INVESTIGATIONS AND OVERSIGHT
SUBCOMMITTEE ON RESEARCH AND TECHNOLOGY

OF THE

COMMITTEE ON SCIENCE, SPACE, AND TECHNOLOGY

OF THE

HOUSE OF REPRESENTATIVES

ONE HUNDRED SEVENTEENTH CONGRESS

SECOND SESSION

MAY 11, 2022

Serial No. 117–56

Printed for the use of the Committee on Science, Space, and Technology

Available via the World Wide Web: http://science.house.gov

U.S. GOVERNMENT PUBLISHING OFFICE
47–494PDF WASHINGTON : 2023

COMMITTEE ON SCIENCE, SPACE, AND TECHNOLOGY

HON. EDDIE BERNICE JOHNSON, Texas, *Chairwoman*

ZOE LOFGREN, California
SUZANNE BONAMICI, Oregon
AMI BERA, California
HALEY STEVENS, Michigan, *Vice Chair*
MIKIE SHERRILL, New Jersey
JAMAAL BOWMAN, New York
MELANIE A. STANSBURY, New Mexico
BRAD SHERMAN, California
ED PERLMUTTER, Colorado
JERRY MCNERNEY, California
PAUL TONKO, New York
BILL FOSTER, Illinois
DONALD NORCROSS, New Jersey
DON BEYER, Virginia
CHARLIE CRIST, Florida
SEAN CASTEN, Illinois
CONOR LAMB, Pennsylvania
DEBORAH ROSS, North Carolina
GWEN MOORE, Wisconsin
DAN KILDEE, Michigan
SUSAN WILD, Pennsylvania
LIZZIE FLETCHER, Texas

FRANK LUCAS, Oklahoma, *Ranking Member*
MO BROOKS, Alabama
BILL POSEY, Florida
RANDY WEBER, Texas
BRIAN BABIN, Texas
ANTHONY GONZALEZ, Ohio
MICHAEL WALTZ, Florida
JAMES R. BAIRD, Indiana
DANIEL WEBSTER, Florida
MIKE GARCIA, California
STEPHANIE I. BICE, Oklahoma
YOUNG KIM, California
RANDY FEENSTRA, Iowa
JAKE LATURNER, Kansas
CARLOS A. GIMENEZ, Florida
JAY OBERNOLTE, California
PETER MEIJER, Michigan
JAKE ELLZEY, TEXAS
MIKE CAREY, OHIO

SUBCOMMITTEE ON INVESTIGATIONS AND OVERSIGHT

HON. BILL FOSTER, Illinois, *Chairman*

ED PERLMUTTER, Colorado
AMI BERA, California
GWEN MOORE, Wisconsin
SEAN CASTEN, Illinois

JAY OBERNOLTE, California, *Ranking Member*
STEPHANIE I. BICE, Oklahoma
MIKE CAREY, OHIO

SUBCOMMITTEE ON RESEARCH AND TECHNOLOGY

HON. HALEY STEVENS, Michigan, *Chairwoman*

MELANIE A. STANSBURY, New Mexico
PAUL TONKO, New York
GWEN MOORE, Wisconsin
SUSAN WILD, Pennsylvania
BILL FOSTER, Illinois
CONOR LAMB, Pennsylvania
DEBORAH ROSS, North Carolina

RANDY FEENSTRA, Iowa, *Ranking Member*
ANTHONY GONZALEZ, Ohio
JAMES R. BAIRD, Indiana
JAKE LATURNER, Kansas
PETER MEIJER, Michigan
JAKE ELLZEY, TEXAS

C O N T E N T S

May 11, 2022

SECURING THE DIGITAL COMMONS: OPEN-SOURCE SOFTWARE CYBERSECURITY

WEDNESDAY, MAY 11, 2022

HOUSE OF REPRESENTATIVES,
SUBCOMMITTEE ON INVESTIGATIONS AND OVERSIGHT,
SUBCOMMITTEE ON RESEARCH AND TECHNOLOGY
COMMITTEE ON SCIENCE, SPACE, AND TECHNOLOGY,
Washington, D.C.

The Subcommittees met, pursuant to notice, at 10 a.m., in room 2318 of the Rayburn House Office Building, Hon. Bill Foster [Chairman of the Subcommittee on Investigations and Oversight] presiding.

U.S. HOUSE OF REPRESENTATIVES
COMMITTEE ON SCIENCE, SPACE, AND TECHNOLOGY
SUBCOMMITTEE ON INVESTIGATIONS AND OVERSIGHT
SUBCOMMITTEE ON RESEARCH AND TECHNOLOGY

HEARING CHARTER

Securing the Digital Commons: Open-Source Software Cybersecurity

Wednesday, May 11, 2022
10:00 a.m. EDT – 12:00 p.m. EDT
Zoom

PURPOSE

The purpose of this hearing is to discuss the unique benefits and risks inherent in open-source software, and to explore the ways in which industry and government can collaborate to enhance open-source cybersecurity. The hearing will examine recent open-source software hacks and subsequent efforts to improve security for the development and deployment of open-sourced software. Members and witnesses will discuss the Federal role in improving open-source cybersecurity, particularly at the National Institute of Standards and Technology (NIST). Finally, the hearing will explore the use and potential misuse of open-source software in the development of critical technologies, including artificial intelligence (AI).

WITNESSES

- **Ms. Lauren Knausenberger**, Chief Information Officer, Department of the Air Force
- **Mr. Brian Behlendorf**, General Manager, Open Source Security Foundation
- **Ms. Amélie Erin Koran**, Non-Resident Senior Fellow, The Atlantic Council
- **Dr. Andrew Lohn**, Senior Fellow, Center for Security and Emerging Technology, Georgetown University

OVERARCHING QUESTIONS

- What are the consequences of insecure open-source software and what are organizations in both the public and private sectors doing to help prevent those consequences?
- What further research and standards activities are needed to secure the open-source software ecosystem of the future?
- What policy changes can help secure the open-source software ecosystem? How can the Federal government, including NIST, most effectively collaborate with industry and other stakeholders to help secure open-source software?

What is Open-Source Software?

The 2019 National Defense Authorization Act defines open-source software as software for which the human-readable source code is available for use, study, re-use, modification,

enhancement, and re-distribution by the users of such software.[1] Simply put, open-source software is code that can be used, modified, and distributed by anyone. The software can be a standalone program, such as the web browser *Firefox* or the operating system *Linux*. It also can be software that serves a specific function as a component of larger programs. As a component open-source software is present in 97% of codebases,[2] meaning that essentially everything done on a computer uses open-source software at some level.

Open-source software's ubiquity is due to the advantages it provides. The open nature of the code allows a broader spectrum of people to both improve and to use the software, and its flexibility ensures that it can be altered to serve the specific needs of the user. It can also provide an alternative to proprietary software—commercial software such as Windows Office. Finally, it is distributed for free, allowing access to technical capabilities users may not otherwise be able to afford.[3]

Users gain access to open-source software and all its attendant benefits through online software repositories. Some organizations, such as the Apache Software Foundation, maintain a select set of internally managed projects each with their own repository.[4] Others, such as GitHub (which is owned by Microsoft), provide a place for anyone to host or maintain a repository and from which anyone can download open-source software.[5] Federal agencies are also producers and distributors of open-source software. A 2016 memo from the Office of Management and Budget (OMB) directed agencies to make 20% of their custom software open source through code.gov, and individual agencies such as NASA and NIST have their own websites to distribute the open-source software they create.[6][7]

Open Source vs. Proprietary Software Security

The risks of open-source software are fundamentally the same as proprietary software: the presence of vulnerabilities can permit the intrusion of bad actors. The difference is in the way vulnerabilities are managed for these two types of software. A long running unsettled debate in the cybersecurity community is whether open-source software is more secure because it has more people evaluating it for bugs, or if proprietary software is more secure because hackers have less access to search for vulnerabilities.[8] Analyses of real-world attack data suggest that open-source software is not less secure on average than proprietary software, and when properly managed, may be more secure.[9]

The Security Challenges of Open-Source Software

One challenge for open-source security is the relative lack of resources dedicated to preemptive security, such as the creation of software that is more secure by design, or the operation of

---

[1] Public Law 115-232. https://www.congress.gov/115/plaws/publ232/PLAW-115publ232.pdf
[2] https://www.synopsys.com/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html
[3] https://www.redhat.com/en/blog/value-open-source
[4] https://www.apache.org/
[5] https://github.com/about
[6] https://obamawhitehouse.archives.gov/sites/default/files/omb/memoranda/2016/m_16_21.pdf
[7] https://code.nasa.gov/
[8] https://dwheeler.com/secure-programs/Secure-Programs-HOWTO/open-source-security.html
[9] https://www.atlanticcouncil.org/resources/breaking-trust-the-dataset/

internal vulnerability checking. Volunteers contribute to open-source projects based on interest, and because of that fewer than 3% of time spent on open-source projects is dedicated to the security of those projects.[10] Identifying which open-source software meets the designation of "critical," and thus is deserving of greater attention to detect vulnerabilities, is an ongoing effort. NIST has issued a definition of critical software, but primarily aimed at government applications. For open-source software the definition of "critical" is largely in the eye of the beholder, though researchers are working now to develop a consensus.

While open-source software has historically been patched quickly following the identification of a vulnerability,[11] the widespread use of the same open-source software by multiple platforms means that a single vulnerability can have a massive impact. Since open-source software is often a component of a larger program, it can also be challenging to determine if a given program is affected by a vulnerability, which in turn makes it hard to know when patching is required. This problem is so prevalent that that 88% of proprietary software contains outdated or otherwise unsafe open-source software.[12]

Noteworthy Open-Source Software Vulnerabilities

On November 24, 2021, a security researcher with Alibaba privately reported to the Apache Software Foundation that their popular Log4j software had a vulnerability which could allow for remote code execution, essentially giving an attacker the ability to run anything—from software that mines cryptocurrency to ransomware—on the underlying server. This vulnerability became public on December 9 and was dubbed Log4Shell.[13] The Log4j software is used to log events by millions of systems, including Amazon Web Services, and one estimate suggested that 10% of all digital assets were vulnerable to it.[14][15] Exploiting Log4Shell is also relatively easy, and proof of concept code was posted on GitHub shortly after the vulnerability was revealed.[16]

The combination of widespread use in software and easy exploitation are why Log4Shell was viewed as a "catastrophic" vulnerability. While the final patch fixing the exploit was issued on December 27, the process of deploying those patches to all affected systems is still ongoing.[17] A full accounting of the damage from hacks exploiting this vulnerability is still unknown.

Another significant open-source vulnerability, called Heartbleed, was discovered in 2014 in the OpenSSL cryptographic software library. OpenSSL provides open-source encryption protocols that are used to protect internet communications, and at the time of the attack at least 66% of all internet sites used servers that relied on these protocols.[18] The bug allowed hackers to bypass encryption to steal information and was relatively simple to exploit.[19] A patch was issued within

[10] *Ibid, page 5*
[11] https://googleprojectzero.blogspot.com/2022/02/a-walk-through-project-zero-metrics.html
[12] https://www.synopsys.com/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html#
[13] https://www.techtarget.com/whatis/feature/Log4j-explained-Everything-you-need-to-know
[14] https://theconversation.com/what-is-log4j-a-cybersecurity-expert-explains-the-latest-internet-vulnerability-how-bad-it-is-and-whats-at-stake-173896
[15] https://www.tenable.com/blog/one-in-10-assets-assessed-are-vulnerable-to-log4shell?utm_source=charge&utm_medium=social&utm_campaign=internal-comms
[16] https://www.tenable.com/blog/cve-2021-44228-cve-2021-45046-cve-2021-4104-frequently-asked-questions-about-log4shell
[17] https://www.zdnet.com/article/log4j-flaw-thousands-of-applications-are-still-vulnerable-warn-security-researchers/
[18] https://heartbleed.com/
[19] https://www.synopsys.com/blogs/software-security/heartbleed-vulnerability-appsec-deep-dive/

a week of discovery and on the same day the vulnerability was made public. However, even years later, hundreds of thousands of devices were still vulnerable to exploitation.[20]

ONGOING ACTIONS TO ADDRESS OPEN-SOURCE SECURITY

Industry Joint Action

In response to the Heartbleed vulnerability, the Linux Foundation organized the Core Infrastructure Initiative (CII) with support from Google, Microsoft, Facebook, and other major technology companies.[21] Over its lifetime CII funded ten grants to pay for security work on critical open-source projects.[22] In 2020, CII transitioned into the Open Source Security Foundation (OpenSSF) with the goal of tackling open-source security more holistically.

The OpenSSF has developed free training courses on secure software development.[23] They also host working groups with the aim of bringing all relevant stakeholders together to work on topics such as best practices for open-source developers.[24] In response to Log4Shell, OpenSSF established an internal project called "Project Alpha-Omega" to directly improve both the most critical open-source software and to create tools that will raise the baseline of security for all open-source software.[25] For example, the first open-source project targeted for security assistance underlies significant parts of most websites.[26]

Repository managers have also taken careful steps to improve security. On February 1, the npm registry that hosts code for JavaScript, a programming language that underpins many internet applications, began requiring two-factor authentication for the top 100 packages. The npm registry intends to roll out two-factor authentication to the top 500 high-impact packages in early 2022.[27] This helps secure these open-source projects from the submission of malicious code through hacked accounts. To secure open-source code itself, GitHub is trialing an AI programming assistant called Copilot that provides live code suggestions as a programmer is working.[28] This program is intended to help programmers produce more secure software.

White House Actions

On May 12, 2021, the Biden Administration released Executive Order 14028, "Improving the Nation's Cybersecurity."[29] Its goal is to address government supply chain security deficiencies in the wake of the SolarWinds hack. While it did not focus specifically on open-source software, much of the guidance produced as a result would affect how software is analyzed, adopted, and secured in Federal systems. For example, the E.O. took several steps to secure Federal software supply chains, from developing security requirements for newly defined "critical software" to

[20] https://blog.malwarebytes.com/exploits-and-vulnerabilities/2019/09/everything-you-need-to-know-about-the-heartbleed-vulnerability/
[21] https://www.coreinfrastructure.org/faq/
[22] https://web.archive.org/web/20190619184614/https://www.coreinfrastructure.org/grants/
[23] https://openssf.org/training/courses/
[24] https://openssf.org/community/openssf-working-groups/
[25] https://openssf.org/press-release/2022/02/01/openssf-announces-the-alpha-omega-project-to-improve-software-supply-chain-security-for-10000-oss-projects/
[26] https://openssf.org/blog/2022/04/18/openssf-selects-node-js-as-initial-project-to-improve-supply-chain-security/
[27] https://github.blog/2021-12-07-enrolling-npm-publishers-enhanced-login-verification-two-factor-authentication-enforcement/
[28] https://copilot.github.com/
[29] https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/

creating guidance for minimum standards for vendors' testing of their software source code. Pursuant to the E.O., NIST is scheduled to publish additional guidelines for periodic review of software supply chain security later this month.

The White House also held a summit on the security of open-source software with industry and government experts on January 13, 2022.[30] The summit explored how to improve the process of identifying and mitigating vulnerabilities in open-source software and shorten response times. This dialogue is ongoing and more updates are expected in the spring of 2022.

Cybersecurity and Infrastructure Security Agency

The Department of Homeland Security's CISA helps Federal civilian agencies, critical infrastructure entities, and the private sector share cybersecurity information and respond to emerging incidents. CISA provides interagency guidance, coordination, and education activities. Launched in 2019, CISA's Information and Communications Technology (ICT) Supply Chain Risk Management Task Force is a public-private partnership created to improve the nation's ability to assess and mitigate threats to the ICT supply chain, including those from open-source software.[31] The Task Force is made up of industry representatives from the information technology and communications sectors as well as Federal partners like NIST. In addition, in February 2022, CISA released a catalogue of free cybersecurity tools developed in partnership with the open-source community.[32]

CISA has also taken over a multi-stakeholder initiative from the National Telecommunications and Information Administration to develop a Software Bill of Materials (SBOM).[33] Modern software products depend on a vast number of components from different developers, code repositories, and other sources. Suppliers of software components also use different naming schemes for the same components. As a result, identifying which vulnerabilities compromise which products can be a challenging technical feat. SBOMs may be able to address this challenge by creating a machine-readable inventory that will enable software developers and users to track software components and dependencies and make responding to vulnerabilities in the event of an incident more straightforward. However, as the Investigations and Oversight Subcommittee heard during its hearing on Supply Chain Security in May 2021, questions remain about the effectiveness of SBOMs as well as the ability of organizations to adopt them.[34]

National Institute of Standards and Technology

NIST is the agency primarily in charge of the nation's cybersecurity standards and best practices. In 2014, pursuant to E.O. 13636, NIST published a voluntary framework for reducing cybersecurity risks to critical infrastructure.[35] NIST has since updated and expanded its guidance to apply to new scenarios, many of which are applicable to open-source software. By statute, Federal agencies must secure their systems according to directives from the Office of

[30] https://www.whitehouse.gov/briefing-room/statements-releases/2022/01/13/readout-of-white-house-meeting-on-software-security/
[31] https://www.cisa.gov/sites/default/files/publications/ict-scrm-task-force_vear-two-report_508.pdf
[32] https://www.cisa.gov/news/2022/02/18/cisa-launches-new-catalog-free-public-and-private-sector-cybersecurity-services
[33] https://www.ntia.gov/SBOM; https://www.cisa.gov/sbom
[34] https://science.house.gov/hearings/solarwinds-and-beyond-improving-the-cybersecurity-of-software-supply-chains
[35] https://www.nist.gov/cyberframework

Management and Budget. Agencies often choose to use NIST's cybersecurity standards and guidelines to protect their non-national security information and communications infrastructure.

NIST has developed several standards and best practices that apply directly to open-source software. NIST offers guidance for organizations to manage the increasing risk of cyber supply chain compromise.[36] Similarly, NIST has produced guidance for vulnerability remediation.[37] The agency has also developed The Secure Software Development Framework (SSDF) to help software developers mitigate vulnerabilities released in software.[38]

E.O. 14028 required NIST to create several new security standards and guidelines for software in Federal systems, including open-source software:

- In June 2021, NIST published a definition of the term "critical software." The Executive Order also directs CISA to develop a list of software categories and products in use or the acquisition process that meet this definition.[39]
- In July 2021, NIST published security guidelines for critical software and minimum standards for vendors' testing of their software source code.[40,41]
- In February 2022, NIST published standards that enhance the security of the software supply chain based on an updated SSDF, including guidance for software developers to provide SBOMs.[42] NIST published revisions to this document on May 5.[43]
- In February 2022, NIST unveiled guidance on practices software producers can undertake to help strengthen the software supply chain.[44]

National Science Foundation

NSF has traditionally played a role in funding open-source software and data repositories across numerous solicitations.[45] NSF is planning to award grants to help secure elements of the open-source ecosystem as part of its new Pathways to Enable Open-Source Ecosystems (POSE) program.[46] The solicitation for proposals closes on May 12, 2022.

ONGOING CHALLENGES

Understanding The Breadth and Depth of the Open-Source Ecosystem

Because of the protean nature of open-source software projects and the myriad developers who contribute to them, a single comprehensive resource tracking all such projects is not feasible. The percentage of code in surveyed codebases that was open source increased from 36% in 2015 to 78% in 2022.[47] In 2018, the Sloan Foundation's Critical Digital Infrastructure Research Fund

[36] https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-161.pdf
[37] https://csrc.nist.gov/publications/detail/sp/800-40/rev-3/final
[38] https://csrc.nist.gov/publications/detail/sp/800-218/final
[39] https://www.nist.gov/itl/executive-order-improving-nations-cybersecurity/critical-software-definition
[40] https://www.nist.gov/itl/executive-order-improving-nations-cybersecurity/recommended-minimum-standard-vendor-or-developer
[41] https://www.nist.gov/itl/executive-order-improving-nations-cybersecurity/security-measures-eo-critical-software-use
[42] https://www.nist.gov/itl/executive-order-improving-nations-cybersecurity/software-supply-chain-security-guidance
[43] https://www.nist.gov/news-events/news/2022/05/nist-updates-cybersecurity-guidance-supply-chain-risk-management
[44] https://www.nist.gov/itl/executive-order-improving-nations-cybersecurity/software-supply-chain-security-guidance
[45] https://www.nsf.gov/awardsearch/showAward?AWD_ID=1348450&HistoricalAwards=false
[46] https://beta.nsf.gov/funding/opportunities/pathways-enable-open-source-ecosystems-pose
[47] *Ibid*

provided funding for thirteen grants to study various aspects of the ecosystem.[48] However, those projects were narrow in scope and did not provide the bird's-eye view of the ecosystem needed to identify crucial open-source software so that resources could be allocated appropriately.

The Linux Foundation has performed a pair of censuses of well-characterized and manageable subsets of the overall ecosystem. The first census, conducted in partnership with the Core Infrastructure Initiative in 2015, sought to identify packages within a particular Linux distribution that were essential to that operating system's security.[49] The second census, completed in March 2022 in partnership with the Harvard Laboratory for Innovation Science, identified the 1,000 most widely deployed open-source libraries in commercial and enterprise applications.[50] This list is meant to help target security efforts towards software with the greatest potential for impactful hacks.

The Federal government also has a role to play in identifying and cataloguing critical software. NIST's publication of a definition of the term "critical software" can assist in identifying the open-source project that most need resources, though NIST's focus is on government software and may not be identical with the software critical to industry.[51] Section 10224 of the *America COMPETES Act of 2022* seeks to address this issue by directing NIST to assess and identify security risks in open-source software.[52]

Resources of the Open-Source Ecosystem Vary Widely

Open-source software is often under-resourced because it lacks the commercial support of proprietary software and because a significant percentage of its developers are volunteers. [53] For instance, prior to the Heartbleed vulnerability, the organization maintaining OpenSSL received an average of just $2,000 per year in donations for their work.[54] With regard to individual developers, a 2020 survey found that 44% received no compensation for their open source work, 49% were compensated by their employer, though many of those also said they worked on additional open-source projects for free. Only 3% were paid by a third party.[55] Focused spending may be needed to encourage work to detect vulnerabilities before they become public. However, it may be challenging to acquire or distribute funds in the volunteer-heavy open-source community.

Industry has attempted to expand the funding dedicated to open-source security in several ways. The Chan Zuckerberg Initiative's Essential Open Source Software for Science program has granted $22.9 million to open-source projects that support biological research since May 2019.[56] In October of 2021 Google announced it was providing $1 million for a pilot program called SOS Rewards to reward improvements in the security of critical open-source software.[57] Google

[48] https://www.fordfoundation.org/campaigns/critical-digital-infrastructure-research/
[49] https://www.coreinfrastructure.org/programs/census-program-i/
[50] https://www.linuxfoundation.org/press-release/the-linux-foundation-and-harvards-lab-for-innovation-science-release-census-of-most-widely-used-open-source-application-libraries/
[51] https://www.nist.gov/system/files/documents/2021/10/13/EO%20Critical%20FINAL.pdf
[52] https://www.congress.gov/bill/117th-congress/house-bill/4521/text/eh
[53] https://xkcd.com/2347
[54] https://www.coreinfrastructure.org/faq/
[55] https://www.linuxfoundation.org/wp-content/uploads/2020FOSSContributorSurveyReport_121020.pdf
[56] https://chanzuckerberg.com/eoss/
[57] https://www.techrepublic.com/article/google-stakes-new-secure-open-source-rewards-program-for-developers-with-1m-seed-money/

also partnered with Microsoft to contribute $5 million to the OpenSSF's Project Alpha-Omega.[58] However, these contributions pale in comparison to the scope of the open-source ecosystem. Moreover, simple infusions of money may not be sufficient to increase security. One of the volunteers working on Log4j said that more funding would have been unlikely to catch the vulnerability, and that instead more involvement by knowledgeable volunteers would go a long way toward increasing the security of the project.[59]

Unique Risks of Open-Source Software for Critical Technologies

Open-source powers technologies ranging from drones to quantum computing to AI. While these applications face similar vulnerabilities to other open-source applications, there are sometimes unique challenges to critical technologies built with open-source software or data. For example, machine learning often requires massive datasets to improve the accuracy of the model. Many organizations or researchers have produced open-source datasets to allow other researchers or developers to train their AI systems.[60] However, malicious actors could theoretically make alterations to these freely available datasets to manipulate an AI system to produce an inaccurate or harmful result. For instance, malicious data might cause an AI to disregard anomalies that would have revealed an intrusion into the system it monitors. One researcher found that poisoning just 0.7% of a dataset was sufficient to bypass defenses.[61]

Another common use of open source within AI development is the sharing of pre-trained models. These models have already been tuned by their developers to produce an intended result, like language processing or image generation. Less technically savvy individuals can then apply the pre-trained model to new tasks or situations.[62] Similar to the issue of poisoned datasets, researchers have found that it is possible to place a backdoor in a model which can provide malicious outcomes only when instructed and is otherwise indistinguishable from a clean model.[63] When the model's code is open source, malicious actors can find weaknesses in the system or create results that were unintended by the developers. For example, when Facebook released a new AI language model called OPT-175B, security researchers were easily able to cause the model to "generate toxic language and reinforce harmful stereotypes."[64] While exploits like these have not yet been detected in the wild, they become more likely as AI systems become more heavily integrated into society.

[58] https://openssf.org/press-release/2022/02/01/openssf-announces-the-alpha-omega-project-to-improve-software-supply-chain-security-for-10000-oss-projects/
[59] https://therecord.media/the-apache-log4j-team-talks-about-the-log4shell-patching-process/
[60] https://cset.georgetown.edu/wp-content/uploads/CSET-Poison-in-the-Well.pdf
[61] https://www.bloomberg.com/opinion/articles/2022-04-24/ai-poisoning-is-the-next-big-risk-in-cybersecurity
[62] https://cset.georgetown.edu/wp-content/uploads/CSET-Poison-in-the-Well.pdf
[63] https://arxiv.org/abs/2204.06974
[64] https://arxiv.org/pdf/2205.01068.pdf

Chairman FOSTER. This hearing will come to order. And without objection, the Chair is authorized to declare recess at any time.

Before I deliver my opening remarks, I wanted to note that the Committee is meeting both in person and virtually. I want to announce a couple of reminders to the Members about the conduct of this hearing. First, Members and staff who are attending in person may choose to be masked, but it is not a requirement. However, any individuals with symptoms, a positive test, or exposure to someone with COVID–19 should wear a mask while present.

Members who are attending virtually should keep their video feed on as long as they are present in the hearing. Members are responsible for their own microphones, and please also keep your microphones muted unless you are speaking. Finally, if Members have documents they wish to submit for the record, please email them to the Committee Clerk, whose email address has been circulated prior to the meeting.

Well, good morning, and welcome to our Members and witnesses. Thank you for joining us for this important hearing on open-source software (OSS) security.

Cybersecurity is certainly an evergreen issue, and today we're focusing on an important and often overlooked corner of the ecosystem. Open-source software is software that's freely available for anyone to use and modify. It's the hidden workhorse of the digital ecosystem, and it's a part of software ranging from standalone browsers to complex commercial operating systems.

It's also common in scientific research. For example, Fermi National Accelerator Lab, where I worked for many years as a physicist, recently announced the development of open-source software to support the control electronics of quantum computers. Even if you're not working with a quantum computer this afternoon, it's safe to say that anyone who has used a computer has relied on open-source software, whether they know it or not.

And yet—and despite its importance, open-source software only seems to draw attention when something goes wrong. In 2014 the Heartbleed vulnerability in OpenSSL prompted a surge of concern and some action to save open-source software as—you know, on the parts of industry and government alike. Good work was done in response to that vulnerability, but interest soon waned, as it often does in Congress and government, and in many ways we find ourselves in the same situation now that we were in back then.

This past winter, the open-source community was once more rocked by a dangerous vulnerability. The Log4j project and its vulnerability, sometimes referred to as Log4Shell, reminded everyone about the dangers of neglecting open-source software. The sheer breadth of organizations affected by that vulnerability in a single piece of software drove home just how much everyone relies on open source.

Open-source generally is an interesting example of the tragedy of the commons. You have the free-rider problem. It's something where normally you would simply say this is a public good and make everyone bear the burden. But—and to—for some packages, which are really relied on universally, then that's probably a good model, but there are difficulties because not everyone benefits equally from special attention being paid to different packages and

so on. So I look forward to trying to get a little deeper in the weeds on how we might decide how to prioritize and where to put our effort and who's to take responsibility for specific packages. It's a tough set of weeds to get into.

This hearing is not meant to look back at the hows and whys of Log4j. Others, including other congressional committees, have already done a good job of that. Instead, this hearing will look forward. It will explore how industry and government can cooperate to make sure open source has resources commensurate with its importance. Those resources are not just financial, but also include technical capabilities, volunteer efforts, and administrative and organizational contributions.

This hearing is also an opportunity to look at some of the dangers of open source that are looming on the horizon. Open-source software is not just in traditional computers. It's in our drones, our AI (artificial intelligence) models, and yes, even our quantum computers. We need to fully understand how open-source resources are used in developing technologies to properly assess the risks that those uses represent.

It is important to remember that no software is ever completely secure. Just as, for instance, you know, Windows and iOS will certainly be hacked many times in the future, there will also be other open-source software vulnerabilities. Rather than seeking perfection, our goal is to structure how we think about open source and how we identify the most critical pieces of open-source software for prioritization, and how we secure that software against intrusion or blunders. If we do that, we will be able to mitigate both the risk of future vulnerabilities and the damage caused when vulnerabilities are exploited.

In a world where our technology so often comes with hidden drawbacks or hidden motivations, open-source software is—often seems to be a charmingly utopian exception. At its best, it is simply goodhearted people creating software out of their own passions and sharing out of a desire for others to benefit from the fruits of that labor. It empowers people of all backgrounds and all levels of technical ability to build upon the work of others and find or make software more suited to their needs. There is something wonderful about that, and I hope that through our conversation with our witnesses here we can contribute to the future of safe and secure open-source software.

[The prepared statement of Chairman Foster follows:]

Good morning, and welcome to our members and witnesses. Thank you for joining us for this important hearing on open-source software security. Cybersecurity is certainly an evergreen issue, and today we're focusing on an important and often overlooked corner of the ecosystem.

Open-source software is software that's freely available for anyone to use or modify. It's the hidden workhorse of the digital ecosystem, and it's a part of software ranging from standalone browsers to complex commercial operating systems.

It's also common in scientific research. For instance, Fermilab—where I worked for many years as a physicist—recently announced the development of open-source software to support the control electronics of quantum computers. Even if you're not working with a quantum computer, it's safe to say that anyone who has used a computer has relied on open-source software, whether they knew it or not.

And yet, despite its importance, open source only draws attention when something goes wrong. In 2014 the Heartbleed vulnerability in OpenSSL prompted a surge of concern and action to save open source on the part of industry and government alike. Good work was done in response to that vulnerability, but interest waned

and, in many ways, we find ourselves in the same situation now that we were in back then.

This past winter, the open source community was once more rocked by a dangerous vulnerability. The Log4j project and its vulnerability, called Log4Shell, reminded everyone of the dangers of neglecting open-source software. The sheer breadth of organizations affected by a vulnerability in a single piece of software drove home just how much everyone relies on open source.

This hearing is not meant to look back at the hows and whys of Log4j—others, including other Congressional committees, have already done an admirable job of that. Instead, this hearing will look forward. We will explore how industry and government can cooperate to make sure open source has resources commensurate with its importance. Those resources are not just financial, but also include technical capabilities, volunteer efforts, and administrative and organizational contributions.

This hearing is also an opportunity to look at some of the dangers of open source that are looming on the horizon. Open-source software is not just in traditional computers; it's in our drones, our AI models, and yes, even quantum computers. We need to fully understand how open-source resources are used in developing technologies to properly assess the risks that those uses represent.

It is important to remember that no software is ever completely secure. Just as, for instance, Windows and iOS will certainly be hacked in the future, there will also be other open-source software vulnerabilities. Rather than seeking perfection, our goal is instead to structure how we think about open source, how we identify the most critical pieces of open-source software, and how we secure that software against intrusion.

If we do that, we will be able to mitigate both the risk of future vulnerabilities and the damage caused when vulnerabilities are exploited.

In a world where our technology so often comes with hidden drawbacks or motivations, open-source software is often a charmingly utopian exception. At its best, it is simply people creating software out of passion, and sharing out of a desire for others to benefit from the fruits of that labor. It empowers people of all backgrounds and levels of technical ability to build upon the work of others and find or make software suited to their needs.

There is something wonderful about that. I hope that through our conversation with our witnesses here today we can contribute to the future of safe and secure open-source software.

Chairman FOSTER. So I have a letter here from GitHub that—which I'd like to enter into the record. And without objection, it is so ordered.

And the Chair will now recognize our Ranking Member and perhaps the other AI programmer in the U.S. Congress, Ranking Member Obernolte.

Mr. OBERNOLTE. Well, thank you very much, Chairman Foster. Thank you for—Chair Stevens, for convening this very important hearing on a topic that is very close to my heart and I think is of critical importance in ensuring the future of cybersecurity of software in the United States.

We all know that open-source software has had a tremendously beneficial impact on software development here in the United States. The very nature of open-source software where we encourage collaboration and reuse of code, I think, enhances the efficiency of software development here, but it also comes with inherent risks. And I think as time goes by and different vulnerabilities are exposed, we're learning more and more about those risks. The fact that the code is out there for anyone to see means that malign actors can look at that code and identify vulnerabilities that they might not have seen if they just knew that a software was operating on the code and didn't have access to the code itself. So we've known about those vulnerabilities.

What we're going to hear about in the hearing today is some other vulnerabilities that have come to light, in particular supply chain vulnerabilities where malign actors might intentionally intro-

duce vulnerabilities in software in the hopes that it's incorporated downstream later when software applications based on those modules are built. That's something that's relatively new in our understanding of open-source software and cybersecurity and something that we certainly need to be on guard against.

We're going to hear from the Air Force on their Project One Platform, which I think is—introduces some really innovative and useful technologies to cope with those vulnerabilities. And also I'm very interested today to hear from some of our experts about a new type of vulnerability that deals with the ubiquity of artificial intelligence now in open-source software.

AI, particularly that based on machine learning, as we know, depends on massive data sets to train AI algorithms. And we're starting to be aware of vulnerabilities that could be caused by manipulation of those data sets. In fact, the statistics are pretty alarming that just changing a small amount of the instances in those data sets can cause very serious problems to occur in the implementation of that artificial intelligence. And a malign actor could introduce an intentional error in the data set that's intended to cause a specific problem.

So these are all things that I'm really happy that we're going to be hearing about in this hearing. And I'm hoping that—as the Chairman also articulated, I'm hoping that we can identify some ways that government can be helpful in solving some of these problems. You know, there is some concern in the open-source community that the heavy hand of government could have a very deleterious effect on the adoption of open-source software. You know, the whole community that has led to open source is about collaboration and transparency. And the moment we take the heavy hand of government down in the form of regulation on that industry I think we run the very risk of causing problems.

But I think we also have a role to play as government. We have vast resources at our disposal. We have amazing people that work in the various branches of our Federal Government and of our NGOs (non-governmental organizations), so we have a lot of talent and tools that we can bring to bear on this problem. So I'm hoping that hearings like this one will be instrumental in catalyzing a very effective and useful adoption of some of the tools the government has to bear on this problem of cybersecurity. And I think it is possible to have a Goldilocks solution where government is here to help and not to hurt.

So with that, I'm looking very much forward to hearing from our witnesses, and I yield back.

[The prepared statement of Mr. Obernolte follows:]

Good morning. Thank you, Chairman Foster and Chairwoman Stevens, for convening this hearing. And thanks to our witnesses for appearing before us today.

We are here today to discuss the benefits and risks of open-source software and to explore the ways that government and industry can work together to improve open- source cybersecurity. I look forward to learning more about the solutions we're trying to catalyze, and the collaborations that are already underway, to solve some of the cybersecurity challenges with open-source software. I'm hopeful that today's hearing will be a productive discussion that will help us learn from the past to improve open- source cybersecurity for the future.

At the risk of oversimplification, open-source software is essentially code that can be used, modified, and distributed by anyone. This code can comprise an entire standalone program, like an open-source web browser or operating system. It can

also comprise a small component or specific function built within a larger standalone program, including proprietary and commercial products. In short, open-source software touches almost every facet of our digital ecosystem.

The ubiquity of open-source software is a function of the benefits and advantages it provides. Its open nature expands the breadth and depth of users that can contribute to, improve, and ultimately use the software. It is also flexible and can be tailored to the specific needs of the end-user without having to reinvent the wheel. Leveraging open- source software can save developers' resources, which can, in turn, be reinvested to foster novel and innovative open-source solutions.

The open nature of open-source, however, is not without inherent risk. Its open and collaborative, community-driven nature means that open-source code can be freely edited or changed. The quality and security of changes or contributions are often dependent upon the governance, structure, and policies of the relevant open-source project or community, which can make it difficult to adequately assess the quality and security of various open-source software.

Understanding when open-source has been modified, what changes have been made, and a method for verification or certification that such changes are sound would go a long way toward improving the overall security of open-source software. I'm particularly excited to learn more about Platform One and the work the Air Force is doing in this space.

The ubiquity of open-source also represents a risk. Since open-source software touches every facet of our digital ecosystem, a security vulnerability in open-source code could have a ripple effect throughout the digital economy if exploited. An example of this is the recent Log4Shell vulnerability in an open-source library. Despite being discovered more than six months ago, efforts are still underway to patch vulnerable systems. One of the pervasive issues that has hindered quick remediation is that it has been difficult to determine where the vulnerable open-source library has been used. It is so embedded in the digital ecosystem that cyber professionals are still uncovering instances of its use.

While a software bill of materials or SBOM-effectively an ingredients list for software- may not have prevented the vulnerability from being written into the open-source code in the first instance, it certainly would go a long way in helping to remediate and patch the issue on the back end. I look forward to hearing more from our experts today on how to employ SBOMs to improve open-source cybersecurity.

Finally, I think that the cybersecurity of open-source software could be improved if we can figure out a method for classifying or categorizing open-source instances that range from the critical to the non-critical.

This would help open-source communities and their contributors to prioritize the most important open-source products for heightened scrutiny. I look forward to hearing more about some of the efforts that the Linus Foundation and OpenSSF have stood up to do just this.

In closing, I think it is important to articulate plainly that open-source security is cybersecurity. Our information and communications infrastructure is only as strong as its weakest link. I'm hopeful that we can have a productive discussion today to put us on the path toward shoring up our digital infrastructure by improving the security of open- source software for the future.

Thank you, Chairman Foster, for convening this hearing. And thanks again to our witnesses for appearing before us today. I look forward to our discussion.I yield back the balance of my time.

Chairman FOSTER. Thank you. And the Chair will now recognize Ms. Stevens for an opening statement.

Ms. STEVENS. Well, good morning and welcome to this joint hearing of the Subcommittee on Research and Technology and the Subcommittee on Investigations and Oversight on securing the digital commons, improving the health of the open-source software ecosystem. What a delightful and exciting topic. I'd like to certainly thank my esteemed colleagues, Chairman and Dr. Foster and Ranking Member Obernolte, for leading this timely and needed hearing.

A supply chain, as they say, is only as strong as its weakest link, and the times when the weakest link happens to be cybersecurity, we see devastating ripple effects and wide-ranging aftershocks. We can no longer operate off of yesterday's mindset and only view sup-

ply chain cybersecurity as an IT (information technology) problem. In order to strengthen America's collective cybersecurity, we must examine all the vulnerable links in the supply chain.

I am deeply proud to be here today to encourage Congress to explore various avenues that government can engage the open-source community to identify and remedy vulnerabilities, the open-source community of which I have hailed from in my previous professional career.

One year ago, President Biden released an Executive order called "Improving the Nation's Cybersecurity." This Executive order tasked the National Institute of Standards and Technology (NIST), a fan favorite of this Committee, to create essential standards for critical software, software supply chain risk management, among other tasks. In the coming days, NIST is expected to publish its final piece of guidance required by the Executive order, but the agency's work to secure the Nation's software is obviously far from finished. One aspect of supply chain security we need to take is an in-depth look at open-source vulnerability landscapes. Many leading companies and organizations don't recognize how many aspects of their critical infrastructure depend on open source.

Open-source software code is available to the public for anyone to use, modify, or inspect. Many elements of NIST's software guidance can be applied to open-source software such as secure software development frameworks. However, they do not address many of the unique challenges inherent in the open-source software ecosystem from inadequate resourcing to vulnerability detection and mitigation. So the point is we don't want to hinder innovation. We just went to do it right, and we don't want to make ourselves vulnerable in the process of innovating. A vibrant open-source ecosystem is an engine to U.S. competitiveness and growth.

The ecosystem benefits Americans every day, including in my home State of Michigan. During the pandemic, open-source applications tracked open hospital beds and helped Michiganders access food for their families when schools were closed. It's the first call I made, open-source platforms responding to the urgency and needs brought on by the COVID–19 pandemic.

But certainly there remains real risk if we leave critical open-source software vulnerable to attack. As both the Heartbleed and Log4j incidents have revealed, open-source software issues can be a threat to our Federal agencies and businesses across the country. There's good work underway but still much more the U.S. scientific enterprise can do to secure open-source software repositories.

Last year, I introduced the *NIST for the Future Act,* which is part of the *America COMPETES Act,* and we will hopefully send that to the President's desk soon. This bill would require NIST to expand its current efforts by assigning severity metrics to vulnerabilities in open-source source software and producing voluntary guidance to help entities that maintain this software to secure it.

The National Science Foundation plays an important role in funding many open-source software and data repositories. NSF has planned to award grants to help secure elements of the open-source ecosystem as part of its new program: Pathways to Enable Open-Source Ecosystems (POSE). Gosh, I wish more people could hear

that because it's also called POSE. I am encouraged by these efforts, which will be further bolstered once we enact and fund the *NSF for the Future Act,* which is also in *COMPETES*.

Securing open-source software is fundamentally a resource problem. I believe the Federal Government can play a role identifying vulnerabilities, providing resources where industry might not, and providing long-term structural security improvements throughout the open-source ecosystem. These efforts are most effective when done in coordination and collaboration with the private sector.

I welcome the recommendations of this expert panel. I again thank my colleagues for convening us and certainly the recommendations on how to improve the coordination between the public and private sector, which are forthcoming. And again, thank you to our witnesses. I yield back, Mr. Chair.

[The prepared statement of Ms. Stevens follows:]

Good morning and welcome to this joint hearing of the Subcommittee on Research and Technology and the Subcommittee on Investigations and Oversight. I would like to thank my esteemed colleagues, Chairman Foster and Ranking Member Obernolte, for leading this timely and needed hearing.

A supply chain is only as strong as its weakest link—and the times when the weakest link happens to be cybersecurity, we see devastating ripple-effects and wide-ranging aftershocks. We can no longer operate off yesterday's mindset and only view supply chain cybersecurity as an IT problem. In order to strengthen America's collective cybersecurity, we must examine all the vulnerable links in the chain. I am proud to be here today to encourage Congress to explore various avenues the government can engage the open-source community to identify and remedy vulnerabilities.

One year ago, President Biden released an Executive Order called "Improving the Nation's Cybersecurity." This executive order tasked the National Institute of Standards and Technology to create essential standards for critical software, software supply chain risk management, among other tasks. In the coming days, NIST is expected to publish its final piece of guidance required by the executive order, but the agency's work to secure the Nation's software is far from finished.

One aspect of supply chain security we need to take an in-depth look at is the open-source vulnerability landscape. Many leading companies and organizations don't recognize how many aspects of their critical infrastructure depend on open source. Open-source software code is available to the public, for anyone to use, modify, or inspect. Many elements of NIST's software guidance can be applied to open-source software, such as the secure software development framework. However, they do not address many of the unique challenges inherent in the open-source software ecosystem, from inadequate resourcing to vulnerability detection and mitigation.

A vibrant open-source ecosystem is an engine for U.S. competitiveness and growth. This ecosystem benefits Americans every day, including in my home state of Michigan. During the pandemic, open-source applications tracked open hospital beds and helped Michiganders access food for their families when schools were closed. But there is real risk if we leave critical open-source software vulnerable to attack. As both the Heartbleed and Log4J (pronounced log-4-J) incidents have revealed, open-source software issues can be a threat to our Federal agencies and businesses across the country.

There is good work underway, but still much more the U.S. scientific enterprise can do to secure open-source software repositories. Last year, I introduced the *NIST for the Future Act,* which is part of the *America COMPETES Act* that we will hopefully send to the President's desk soon. This bill would require NIST to expand its current efforts by assigning severity metrics to vulnerabilities in open-source software and producing voluntary guidance to help entities that maintain this software to secure it.

The National Science Foundation has played an important role in funding many open-source software and data repositories. NSF is planning to award grants to help secure elements of the open-source ecosystem as part of its new program "Pathways to Enable Open-Source Ecosystems," or POSE. I am encouraged by these efforts, which will be further bolstered once we enact and fund the *NSF for the Future Act* that is also in *COMPETES*.

Securing open-source software is fundamentally a resource problem. I believe the Federal government can play a role identifying vulnerabilities, providing resources

where industry might not, and driving long-term structural security improvements throughout the open-source ecosystem. These efforts are most effective when done in coordination and collaboration with the private sector.

I welcome the recommendations of this expert panel on how to improve the coordination between the public and private sector on securing the open-source ecosystem, and any additional recommendations you may have for this Committee to consider.

I want to again thank the witnesses for being here today to help us tackle these challenging issues. I yield back.

Chairman FOSTER. Thank you. And the Chair will now recognize Mr. Feenstra for an opening statement.

Mr. FEENSTRA. Thank you, Chairman Foster and Chairwoman Stevens. Thank you for your passion. And, Ranking Member Obernolte, thank you for being here today at this hearing. I also want to thank our expert witnesses for participating today. I look forward to the discussion and learning more about ways to improve open-source software security.

Open-source software is a key component to modern software development. Over the past 2 decades, open-source software has become widely adopted and has a vast number of applications from powering our small personal devices to our supercomputers. Open-source software is largely created by volunteers on their own time who often do not receive any sort of compensation for their work, but rather work on projects that they are passionate about that may be useful to others. It is collaborative in nature, as it is available for anyone to use, modify, and share for best user ability and accessibility. Additionally, open-source software is often available free of charge, which allows users to have access to technological capabilities that they may not be able to otherwise.

While open-source software offers many benefits, there are also risks involved in using this type of software. One of the main challenges of open-source software is the lack of dedicated resources for security and internal vulnerability checks. If open-source software has a security vulnerability, it could cause widespread harm to all users. What's more, because open-source software is typically part of another software component, it may be tough to determine when and where a patch may be needed.

Critical technologies such as artificial intelligence often have their own unique challenges when it comes to open-source software security. For example, large datasets are used to train artificial intelligence systems to improve their accuracy. If malicious actors manipulate or poison these datasets, the models will be corrupted and could produce inaccurate and harmful outcomes.

Federal science agencies are actively working to address some of the ongoing challenges to open-source software security. The National Institute of Standards and Technology, NIST, has developed standards and best practices that apply to open-source software. NIST also produced guidance for managing compromised cyber supply chains and fixing vulnerabilities.

On May 12, 2021, the President issued an Executive order on "Improving the Nation's Cybersecurity" to enhance the security and integrity of the software supply chain. The Executive order required NIST to create new security standards for software, including open-source software.

The National Science Foundation (NSF) also recently launched a new program called Pathways to Enable Open-Source Ecosystems,

POSE, to harness the power of open-source development for the creation of new technology solutions. Additionally, many NSF-funded research projects produce open-source software, hardware, or data platforms that perform further innovation. It is important that security risks to the open-source ecosystem are adequately addressed and that the necessary resources are dedicated to bolstering cybersecurity.

Improving our Nation's cybersecurity is particularly important to me. My district has recently been targeted by malicious cyberattacks to our agriculture supply chain.

I hope we can have a productive discussion today about improving security in open-source software without compromising its benefits. I once again want to thank the witnesses for being here to discuss this important topic, and I look forward to hearing your solutions. Thank you, and I yield back.

[The prepared statement of Mr. Feenstra follows:]

Thank you, Chairman Foster and Chairwoman Stevens for holding today's hearing. And thank you to our expert witnesses for your participation here today. I look forward to the discussion and learning more about ways to improve open-source software security.

Open-source software is a key component of modern software development. Over the past two decades, open-source software has become widely adopted, and has a vast number of applications from powering small personal devices to supercomputers.

Open-source software is largely created by volunteers on their own time who often do not receive any sort of compensation for their work, but rather work on projects they are passionate about that may be useful to others.

It is collaborative in nature, as it is available for anyone to use, modify, and share for better usability and accessibility.

Additionally, open-source software is often available free of charge, which allows users to have access to technological capabilities that they may not be able to otherwise.

While open-source software offers many benefits, there are also risks involved in using this type of software. One of the main challenges of open-source software is the lack of dedicated resources for security and internal vulnerability checks. If open-source software has a security vulnerability, it could cause widespread harm to all users.

What's more, because open-source software is typically part of another software component, it may be tough to determine when and where patching may be needed.

Critical technologies such as artificial intelligence often have their own unique challenges when it comes to open-source software security. For example, large datasets are used to train artificial intelligence systems to improve their accuracy. If malicious actors manipulate or poison these datasets the models will be corrupted and could produce inaccurate or harmful outcomes.

Federal science agencies are actively working to address some of the ongoing challenges to open-source software security. The National Institute of Standards and Technology (NIST) has developed standards and best practices that apply to open-source software. NIST also produced guidance for managing compromised cyber supply chains and fixing vulnerabilities.

On May 12, 2021, the President issued an Executive Order on "Improving the Nation's Cybersecurity" to enhance the security and integrity of the software supply chain. This Executive Order required NIST to create new security standards for software, including open-source software.

The National Science Foundation (NSF) also recently launched a new program called "Pathways to Enable Open-Source Ecosystems" (POSE) to harness the power of open- source development for the creation of new technology solutions. Additionally, many NSF-funded research projects produce open-source software, hardware, or data platforms that promote further innovation.

It is important that security risks to the open-source ecosystem are adequately addressed and that the necessary resources are dedicated to bolstering cybersecurity.

Improving our nation's cybersecurity is particularly important to me, as my district has recently been targeted by malicious cyberattacks to our agriculture supply chain.

I hope we can have a productive discussion today about improving security in open- source software without compromising its benefits. I once again want to thank our witnesses for being here to discuss this important topic, and I look forward to hearing your solutions.

I yield back.

Chairman FOSTER. Thank you. And if there are other Members who wish to submit additional opening statements, your statements will be added to the record at this point.

[The prepared statement of Chairwoman Johnson follows:]

Good morning. Thank you everyone for joining us for this joint Subcommittee hearing. I especially want to thank Chairs Foster and Stevens, as well as Ranking Members Obernolte and Feenstra, for their leadership on the important issue of open-source software cybersecurity.

Cybersecurity is a perennial problem. It is one we have frequently examined here in the Science Committee. Nearly one year ago, we held a hearing on ways to improve the cybersecurity of software supply chains. Our expert witnesses spoke of a need to improve the security of open-source software to protect the entire software supply chain.

Their foresight was astute. At the end of last year, a vulnerability called Log4Shell was found in a piece of crucial and widely used open-source software. Thousands of organizations and systems were affected, and the work of protecting those systems is still ongoing. One leading cyber company called this software exploit "the single biggest, most critical vulnerability ever." It is clear that we must dedicate more resources to securing open-source software.

Our government agencies have been working hard to support this goal. NIST, in particular, has released extensive guidance for the successful development of secure software. An executive order from last May pushed the agency to do even more. They have released a definition of critical software that can guide the focus to the most important pieces of open-source software. And just last week, NIST issued updated guidance on supply chain risk management, completing a two-and-a-half-year process for how best to handle software in the supply chain.

But NIST cannot solve this problem alone. This is a key moment for government to partner with industry. Our expert witnesses can provide perspectives on open source informed by their time spent working for industry, non-profits, the military, and the civilian government. Their insights will help us understand both the technical challenges and the underlying culture of the open-source community.

Armed with that understanding, we can steer resources towards where they will do the most good. We can also map out the complex ecosystem of those who produce open-source software, and provide training and other resources to help make it secure. We can find more ways for agencies like NIST to collaborate with industry experts and other folks developing and maintaining open-source software across the country.

We will also look to the future. Open source is a critical part of many developing technologies. It enables the growth of artificial intelligence and makes the technology accessible to a wider range of people. Yet the dangers posed by open-source software exist here as well. Bad actors will inevitably try to manipulate open-source datasets to control AI. This is a frightening possibility as AI becomes a bigger part of all our lives.

The risks of open source should not outweigh its benefits. Properly resourced and made secure, open-source software can do a lot of good for a lot of people.

I welcome the recommendations of our expert panel to guide us in that goal. Thank you, and with that I yield back.

Chairman FOSTER. And at this time I'd like to introduce our witnesses. Our first witness is Ms. Lauren Knausenberger. Ms. Knausenberger is the Chief Information Officer (CIO) of the Department of the Air Force comprised of the Air Force and Space Force. Ms. Knausenberger leads two directorates and supports 20,000 cyber operations and support personnel around the globe. She provides oversight of the Air Force's information technology portfolio, including the information technology investment strategy

from networks to cloud computing. Prior to joining the Air Force she was a founder and President of a consulting firm specializing in commercial technologies that could be applied to the government's missions.

After Ms. Knausenberger is Mr. Brian Behlendorf. Mr. Behlendorf is the General Manager of the Open Source Security Foundation, a project hosted by the Linux Foundation with a goal of securing the open-source ecosystem. He has served as an advisor to—an open source to the U.S. Department of Health and Human Services (HHS) and the Office of Science and Technology Policy (OSTP). Mr. Behlendorf was a founding volunteer President of the Apache Software Foundation and serves on the Board of Directors of the Mozilla Foundation and the Electronic Frontier Foundation.

Our third witness is Ms. Amélie Koran. Ms. Koran is a nonresident Senior Fellow with the Atlantic Council's Cyber Statecraft Initiative. During her 30-year career, she has supported work across government agencies, including the U.S. Department of the Interior, the Treasury Department, and the Office of the Inspector General (OIG) within the Department of Health and Human Services. In 2014 she was detailed to the Executive Office of the President to support the Federal CIO in reviewing cybersecurity legislation. She was one of the original cofounders of the U.S. Digital Service and part of the Presidential Management Council's Rotation Program.

And our final witness is Dr. Andrew Lohn. Mr. Lohn is a Senior Fellow at Georgetown Center for Security and Emerging Technology, or CSET, where he works on the CyberAI Project. Prior to joining CSET, he was an Information Scientist at the RAND Corporation where he led research focusing mainly on cybersecurity and artificial intelligence. Andrew has also worked in materials science and nanotechnology at Sandia National Labs, NASA (National Aeronautics and Space Administration), and Hewlett-Packard Labs. He's published in a variety of fields, and his work has been covered in the *MIT Technology Review* and by the BBC (British Broadcasting Corporation).

And, as our witnesses should know, you will each have five minutes for your spoken testimony. Your written testimony will be included in its entirety in the record for the hearing. When you have all completed your spoken testimony, we will begin with questions. Each Member will then have five minutes to question the panel.

And so now we will start with Ms. Knausenberger.

TESTIMONY OF MS. LAUREN KNAUSENBERGER,
CHIEF INFORMATION OFFICER,
DEPARTMENT OF THE AIR FORCE

Ms. KNAUSENBERGER. Good morning, Chairman Foster, Chairwoman Stevens, Ranking Member Obernolte, Ranking Member Feenstra, and distinguished Members of these Committees. Thank you for inviting me today to discuss the benefits of open source and how we can work together as a whole of society to enhance open-source cybersecurity.

As an aside, I will share that our software development community was very energized to hear that we have Members who can write AI algorithms and create e-sports games, as well as a collec-

tion of technologists and enthusiasts who are asking about this topic and raising it to the level of national attention, so thank you for that.

I will share that a few weeks ago I was impressed and perhaps a bit humbled seeing the way that Starlink handled the communication issues in the Ukraine and specifically that they were able to defeat Russian jamming in Ukraine to solve problems with code and to push capability halfway around the world to keep the Ukrainians connected. And these were code pushes that were done in days. If we were looking at this in a military context and trying to bring code around the world, to push it to a pristine disconnected weapons system, it often would take us much longer to do this. Now, that same level of speed that we saw with Starlink or better is needed to protect our country from emerging threats like hypersonic missiles and to ensure that we can stay ahead of the actions of our adversaries and in lockstep with our allies.

It is entirely possible that a future conflict to preserve our way of life is decided by features, fixes, and updates to software-intensive systems that must take place in minutes or hours. And this means that we must learn quickly as a department and leverage the knowledge and best practices of the entire development community.

Now, first, I want to share that I personally am very bullish on open-source software. It's an incredible community of people, as a few of you have mentioned in your opening statements, that want to drive maximum benefit for all. And the top developers and companies in the world are using it and contributing back to open source. And those companies include Google, Microsoft, Red Hat, and Intel as the world's top four contributors to open source. And if you go and speak with those companies and the developers in those companies, they are actually spending a good percentage of their time contributing back to open source that they use because they want to make sure that it is maintained and that it is secure and that they can leverage it for the commercial capabilities that they layer on top of that open-source technology.

It's transparent. You can see the code base. You can even see how the developers go through thinking about how they'll fix a particular bug and the online dialog around fixing a particular—adding a particular feature. It is often more secure in my opinion because it is thoroughly reviewed and vetted by the community, as well as battle-tested by companies around the world. And when there is an issue, it's fixed quickly and openly.

If I compare and contrast SolarWinds and Log4j, Log4j, it was found pretty quickly by the community. It was fixed pretty quickly. The whole process was very transparent. There was open dialog. If we look at SolarWinds, it took us a while to figure out that there was a problem. There were multiple steps, and it did take longer to push those fixes.

Open source allows us to keep costs down by allowing reuse of very good code, and it avoids vendor lock, while allowing the best developers who want to work with us to partner with us. I'll also posit that open-source software has come a long way, especially over the last few years, and I can give credit somewhat to the open-source—the OpenSSF, as well as the Linux Foundation for their

focus on it, as well as just general awareness among the software community about the importance of cybersecurity, as well as the incredible visibility given to vulnerabilities in our national press as well.

Now, while I see the benefits of open source, we in the Department of Defense, we do need to doubly ensure the integrity of the code that we use, as well as ensure that we provide valuable contributions back to the open-source community without giving away protected information critical to our competitive advantage. And those two points were outlined in a recent memo by John Sherman as we put more on the record that we as the Department of Defense are embracing open-source software in policy.

We do see it is our responsibility to independently scan and test all code that we use, whether that is commercial code, whether that is open-source code, and we take this very seriously in the Department, that we maintain awareness of what software is on our networks, that we scan that software, and that we update it to the best of our ability.

Now, historically, our development teams had to do this independently. An individual development team would pull code from an open-source repository or code that was developed by that team, it would go through the pipeline, and that team would be independently responsible for ensuring that that code was updated and secure. We launched Iron Bank as part of Platform One to help our development teams to do this more efficiently and more consistently across the entire department, as well as to open it up to the broader community. And we have had at least one commercial bank pulling containers from Iron Bank, as well as some Fortune 500 players. The intent there is to make sure that the code is secure and current and containerized, that it is accredited for use, which is often a challenge within the Department of Defense, and that it is available to that community.

And further, even after we've done all of these checks within our development environments, we still have a pretty active program for bug bounties, for vulnerability disclosures, and active hacking events where we have on multiple occasions had hackers come in to do a gray hat or white hat review of our systems. They have found vulnerabilities in open-source software, and they have contributed back to the community. We've also had instances where we found dependencies that were previously unknown and contributed back because we are putting a lot of rigor into the checks that we go through. And I see this as really one of the big benefits that we as the Department of Defense bring to the open-source community.

So in our business the adversary consistently gets a vote. It's not just about market share for us. It's about winning. It's about maintaining our competitive advantage as a nation, and it's about ensuring our way of life. We must drive the time and tempo to deliver the capabilities that we need to win, and I thank you again for the opportunity to testify this morning. I welcome your questions.

[The prepared statement of Ms. Knausenberger follows:]

NOT FOR PUBLICATION UNTIL RELEASED BY
HOUSE SCIENCE, SPACE, AND TECHNOLOGY COMMITTEE
SUBCOMMITTEE ON INVESTIGATIONS AND OVERSIGHT
SUBCOMMITTEE ON RESEARCH AND TECHNOLOGY
UNITED STATES HOUSE OF REPRESENTATIVES

DEPARTMENT OF THE AIR FORCE PRESENTATION TO THE
HOUSE COMMITTEE ON SCIENCE, SPACE, AND TECHNOLOGY
SUBCOMMITTEE ON INVESTIGATIONS AND OVERSIGHT, AND
SUBCOMMITTEE ON RESEARCH AND TECHNOLOGY
UNITED STATES HOUSE OF REPRESENTATIVES

HEARING DATE/TIME: May 11, 2022 10:00 A.M.

SUBJECT: Securing the Digital Commons: Open-Source Software Cybersecurity

STATEMENT OF:

Ms. Lauren Knausenberger
Chief Information Officer
Office of the Secretary of the Air Force

NOT FOR PUBLICATION UNTIL RELEASED BY
HOUSE SCIENCE, SPACE, AND TECHNOLOGY COMMITTEE
SUBCOMMITTEE ON INVESTIGATIONS AND OVERSIGHT
SUBCOMMITTEE ON RESEARCH AND TECHNOLOGY
UNITED STATES HOUSE OF REPRESENTATIVES

Introduction & Strategic Implications

Less than five years passed from the formal establishment of the Manhattan Project to the operational use of a nuclear weapon, at the time an unprecedented acceleration of research, development, and operationalization of a new capability. A future conflict may be decided by features, updates, or fixes to software-intensive systems that take place in hours or even minutes, not days, much less years. In order to meet the strategic imperative this moment requires, the Air Force must lower the barrier to fielding new algorithms and systems while increasing the utilization of commercial software, including open-source investments, to diversify the innovation space for the Department.

Open-source software (OSS) is software that has the source code made publicly available, is licensed for broad public reuse, and is typically distributed free of charge. By leveraging open-source software, the development of large-scale software projects can be drastically accelerated due to the re-use of previously developed code and the open systems architectures that open-source software tends to drive. In order to take maximum advantage of the opportunities provided by open-source software, entities in charge of software development efforts should take steps to mitigate some of the risks associated with open-source software.

OSS should not be confused with so-called "shareware," "freeware," or "freemium" software products that are distributed free of charge but require the user to pay to unlock some additional functionality or to continue using the software after a trial period. That's not to say that there are not successful business models built around open-source software: any software requires support and maintenance, and some companies have made a robust business in providing support services for free and open-source software. As an example, Red Hat Enterprise Linux (RHEL) is a free and open-source product for which Red Hat provides paid, enterprise-class support.

Background

Software Life Cycle – Development, Distribution, and Maintenance

The life cycle of an open-source software product can broadly be divided into development, distribution, and maintenance. Development refers to the specification of what the software should do and writing the code to implement the desired functionality; distribution refers to the process of packaging the software appropriately and providing it to the end users (which may be software developers in the case of software dependencies); and maintenance refers to the process of fixing errors (frequently referred to as "bugs"), updating software to maintain compatibility with other software and hardware systems, and incorporating any updates to the dependencies for that software. Due to the constantly evolving nature of modern agile software these life-cycle phases frequently overlap significantly, with features regularly being added to software that is continuously distributed and maintained.

Software Dependencies

A piece of software that is incorporated into a larger piece of software is referred to as a *dependency* of the larger piece of software. Each dependency may in turn have other dependencies (which may in turn have dependencies...) The total collection of dependencies and dependencies-of-dependencies is referred to as the set of *transitive dependencies*, and large software projects may have thousands of transitive dependencies.

Open-source software may be distributed on its own (as is the case with the Android Open Source Project (AOSP, commonly just "Android") mobile operating system that powers Android smartphones) or as part of a proprietary product (Apple's proprietary iOS mobile operating system that powers the iPhone

platform is based on the open-source BSD operating system). In that case, the open-source software is a dependency of the commercial product.

Because of the way that software is constructed out of many layers of dependencies, even if only proprietary software is purchased there are likely still open-source software dependencies and therefore open-source software supply chains that must be considered.

Software Risks

Incorporating open-source software into a software project can significantly speed up the delivery and decrease the cost of developing that software due to re-use of previously developed software. It would be difficult to name a successful software project today that didn't rely on OSS in some form.

Successfully using open-source software requires being cognizant of certain risks:

- The incorporation of any software dependencies (whether open- or closed-source) into your project can expose you to responsibility for maintaining that dependency during the lifetime of your project. If the company or team that developed the software decides to end support for the product, then users may be forced to find a replacement product or pay for an expensive one-off support contract for maintenance. This maintenance also includes the correction of security deficiencies as they are discovered; for this reason, the use of unmaintained software poses both a business and cyber security risk.
    - Example: For years after Windows XP was formally discontinued, Microsoft continued to support existing Windows XP customers via support contracts that increased in price over time (due to a smaller and smaller customer base to distribute the cost across).
- Any software dependency (open-source or proprietary) may inadvertently introduce security flaws into your software product. These broadly fall into the categories of known flaws (these are typically published as Common Vulnerabilities and Exposures, or CVEs) and as-yet-undiscovered flaws (known as "zero-days" in the cybersecurity community). Security-conscious organizations such as the DoD may have more stringent requirements for mitigating these flaws than other organizations, making it difficult for them to utilize the dependency because of a lack of evidence supporting the security of the dependency, disagreement over the severity of the flaws in the software, or timelines for mitigations of flaws that don't satisfy policy or operational requirements.
    - Example: Several months ago, several severe zero-day vulnerabilities in an open-source logging framework named "log4j" were discovered and eventually disclosed as CVEs; thousands or millions of software projects needed to incorporate newer versions of log4j into their projects quite rapidly in order to avoid falling victim to cyber attacks.
- The distribution channels for software may themselves come under attack – this is what is commonly referred to as a "supply chain attack." These sorts of attacks come in several forms. A malicious developer may decide to publish a version of the software that is intentionally flawed, and users that automatically update to the latest version may suffer a degradation in the capability of their system as a result. Malicious developers may publish copy-cat software using a similar name to a common piece of software that is malicious software, hoping that a user will accidentally download and use their malicious software. Finally, if a distribution channel is insufficiently hardened against attack, an adversary may be able to make changes to software in the distribution channel without the developer or using realizing it.
    - Example: The author of a popular open-source package named "left-pad" grew disgruntled with the open-source community and intentionally deleted the software from

the distribution service, briefly causing outages of several prominent websites and probably thousands of less prominent ones.

- Example: The internal development environment at SolarWinds was compromised in a sophisticated supply chain attack that introduced malicious code into their proprietary (i.e. not OSS) software product, which in turn exposed the networks of dozens of governmental and commercial entities to attack.

Software Policy

The documents listed below are a few of the governing policies that relate to the secure use of open-source software by the Air Force. They are not intended to be comprehensive.

Executive Order 14028, Improving the Nation's Cybersecurity (May 12, 2021)

This executive order mandates several efforts to improve the cybersecurity posture of the country. Of particular note to the topic of software dependencies is section 4, "Enhancing Software Supply Chain Security." This section mandates, among other things, the use of secure build environments for compiling software source code into usable programs and the creation of a Software Bill of Materials (SBOM) that is distributed with software that is utilized by the Federal government. This SBOM will assist in knowing what software dependencies exist within a project in order to accurately assess the overall risk of using that project, and to aid in the long-term maintenance and operations of the software. In the log4j example above, the Air Force's use of SBOMs, described below, allowed for the rapid identification of systems that required an updated version of log4j.

DoD CIO Memo on Software Development and Open-Source Software (Jan 24, 2022)

This memorandum expands upon and clarifies a 2009 memo on open-source software use in the DoD. Key points included are that the use of open-source software is explicitly allowed in the DoD, that DoD personnel may contribute code to open-source software as part of their regular duties, and that the DoD must actively manage software supply-chain risk in accordance with the 2018 DoD Cyber Strategy.

DoD Development, Security, and Operations (DevSecOps) Reference Design

The DevSecOps reference design outlines the technology, people, and procedures necessary to create a secure environment for developing and operating secure software. Software development organizations within the Air Force look to the reference design as a starting point for standing up a software development center.

Platform One Efforts to Enhance Software Supply Chain Security

Platform One is an Air Force organization that serves as enterprise provider of development, security, and operations tools and services for the DoD. These services include a secure gateway between secure government cloud environments and the commercial internet; a managed platform-as-a-service for developing and deploying cloud-based applications; the Iron Bank hardened container repository, and the Big Bang implementation of the DoD DevSecOps reference design.

Iron Bank (DoD Hardened Container Repository)

Platform One's Iron Bank is a repository for container images, which is one of the most popular software packaging formats for both OSS and proprietary software that is designed to run in the cloud. Containers are a "cloud native" method of packaging containers and are widely used at Google, Amazon, Microsoft,

Oracle, and virtually every organization that provides or hosts cloud-based software, including Platform One. Iron Bank adds several levels of security that today mitigate some of the risks mentioned above:

- When projects are onboarded, developer identities are verified, reducing the risk of a malicious developer intentionally inserting malicious code into a codebase.
- Copy-cat project names that might confuse users or developers into using the wrong project are not allowed.
- Automated security checks and validation by cybersecurity experts enforce best practices in the development of software containers, potentially reducing the severity of any "zero-day" vulnerabilities that may exist inside of that container.
- Software is compiled and packaged using secure "pipelines" that run inside of the government-controlled software environment, protecting against the introduction of malicious code during the packaging process.
- Software Bills of Materials (SBOMs) are generated for every container in the repository, allowing organizations using those containers to build an accurate, up-to-date inventory of software (including all the dependencies) running in their environment.
- Software in the repository is continuously scanned and compared to the most up-to-date list of Common Vulnerabilities and Exposures (CVEs), so that when a vulnerability is discovered it can be remediated in a timely fashion.
- The Iron Bank repository is hosted in a secure cloud environment and is continuously monitored and assessed according to DoD cyber security standards, ensuring the integrity and availability of the repository and securing it from attack by our adversaries.

Iron Bank has been a successful venture to date: there are approximately 1,000 containers available on Iron Bank, each of which has been assessed against DoD criteria for hardened containers. In the near term, Platform One is focused on improvements to allow Iron Bank to scale in an agile fashion. These include publishing fully automated assessments of a container image to provide fast, objective feedback to DoD organizations that are considering using a piece of software from Iron Bank (a "nutrition label" for a software container), increased use of digital signatures to validate the entire software supply chain, and automated notifications to users (on an opt-in basis) to notify them if they may be using a piece of software that has had a vulnerability disclosed. Because it is a centralized repository, Platform One can search across the repository for critical vulnerabilities (such as the log4j vulnerabilities) and notify anyone that is impacted by the vulnerability (this search is enabled by SBOMs, which provide an inventory of the software in a system). On occasion, the Iron Bank team has provided the first notification to OSS developers that one of their dependencies is vulnerable and needs to be updated.

Big Bang (DevSecOps Reference Design Implementation)

Secure software operations depend on both secure software and correct configuration of that software. Platform One provides a secure baseline for a Development, Security, and Operations (DevSecOps) under the name Big Bang. By adhering to the reference design, Big Bang provides compartmentalization of running applications to minimize the impact of any security incident, implements the services and connections to conduct active cyber defense, and runtime security to detect and neutralize threats to the applications. Big Bang also increases developer productivity and return-on-investment by providing a common baseline for developing and deploying applications. Big Bang is itself an open-source piece of software. Iron Bank (and other Platform One services) utilize Big Bang to provide a secure environment for hosting their applications.

Conclusion

Iron Bank is a successful effort to reduce some aspects of supply chain risk for both open-source and proprietary/commercial software for DoD use. Software security (like cyber security in general) is a constantly evolving and constantly improving effort, not a line of effort that can be "completed" and then focus turned elsewhere. Iron Bank will continue to evolve as better techniques for avoiding software supply-chain risk are developed and implemented. By presenting an enterprise capability for consuming containerized software, Iron Bank users can avoid "re-inventing the wheel" and unnecessarily duplicating the work that has already been done. Finally, Iron Bank and Big Bang serve as foundational capabilities for building enterprise development, security, and operations solutions.

LAUREN BARRETT KNAUSENBERGER

Lauren Barrett Knausenberger is the Chief Information Officer for the Department of the Air Force, comprised of the U.S. Air Force and U.S. Space Force. Ms. Knausenberger leads two directorates and supports 20,000 cyber operations and support personnel around the globe with a portfolio valued at $17 billion. She provides oversight of the Air Force's Information Technology portfolio including the Information Technology investment strategy from networks to cloud computing, Enterprise policies, information resources management, IT innovation initiatives, information assurance, and related matters for the Department of the Air Force. As Chief Information Officer (CIO), Ms. Knausenberger delivers cyber security, and enforces Freedom of Information Act and Privacy Act laws. She integrates Air Force warfighting and mission support capabilities by networking and securing air, space, and terrestrial assets. Ms. Knausenberger also leads career management initiatives for 10,000 IT/Cyber civilian personnel across all human resources facets from recruiting to professional development.

In 2017, Ms. Knausenberger joined the U.S. Air Force to drive innovation across the Department of Defense, speed adoption of emerging technologies, and create stronger partnerships between DoD, start-ups, and the venture community as the Chief Transformation Officer and Director of Cyberspace Innovation under the Deputy Chief Information Officer. Prior to joining the Air Force, she was the founder and President of Accellint Inc., a consulting firm and Venture Partner with NextGen Angels, specializing in solving problems of national security importance, and investing in commercial technologies that could be applied to a government mission. As CIO, she continues to go after hard problems, calculates risk differently, and ensures to keep a constant pulse on the voice of the Airmen.

Prior to her time as an entrepreneur and venture capitalist, Ms. Knausenberger held positions of increasing responsibility at American Management Systems and CACI, beginning as a systems analyst and designer, then as a go-to project manager for program turnarounds, a program manager for large IT and finance programs, and finally as a division manager overseeing much of the company's Intelligence Community portfolio. During this time, Ms. Knausenberger had the pleasure of serving the mission of the CIA directly as well as leading and mentoring cross-functional teams and organizations supporting national security missions.

Ms. Knausenberger earned a Bachelor of Science degree in decision and information sciences from University of Maryland and a Master of Business Administration from University of Pennsylvania Wharton School of Business.

EDUCATION
2003 Bachelor of Science, Decision and Information Sciences, University of Maryland-R.H. Smith School of Business, College Park, Md.
2012 Master of Business Administration, University of Pennsylvania-Wharton School of Business, Philadelphia, Pa.

CAREER CHRONOLOGY
1. June 1998–August 1999, Gifted and Talented Program (Cryptography/Intelligence Analyst Intern), National Security Agency, Fort Meade, Md.
2. July 2003–July 2004, Business Systems Analyst, American Management Systems, Fairfax, Va.
3. July 2004–October 2015, Division Manager-Program Director for Large IT Programs and Intelligence Community Finance, CACI, Fairfax, Va.
4. October 2015–June 2017 President, Accellint, Inc., Arlington, Va.
5. June 2017–August 2020, Chief Transformation Officer and Director of Cyberspace Innovation, the Pentagon, Arlington, Va.
6. August 2020–present, Deputy Chief Information Officer, the Pentagon, Arlington, Va.
7. February 2021–present, Chief Information Officer, the Pentagon, Arlington, Va.

AWARDS AND HONORS
FedScoop Top Women in Technology
Jimmy Doolittle Fellowship Award
Wash100 Award

(Current as of February 2021)

Chairman FOSTER. Thank you. And next is Mr. Behlendorf.

TESTIMONY OF MR. BRIAN BEHLENDORF, GENERAL MANAGER, OPEN SOURCE SECURITY FOUNDATION

Mr. BEHLENDORF. Chairman Foster, Chairwoman Stevens, Ranking Members Obernolte and Feenstra, and Members of the Subcommittee, thank you for the invitation to speak today.

Open-source software is deeply embedded inside every software product, digital platform, smart device, and industrial machinery we have. Our markets, our energy grid, our transport systems, all the conveniences of the modern world would not function without open-source software. Open source comes to us not exclusively or even primarily from a small number of large companies as we might think but from a constellation of different organizations and individuals whose collective efforts are combined and remixed by the thousands into the products and services we see as consumers.

Let's pause for a moment and note just how awesome it is, given so many of the problems we have in this world when we try to work together, that this kind of decentralized global effort is able to produce anything useful at all, let alone the bedrock foundations for digital products and services and upon which we live our lives.

And yet for all these different components from different teams and sources, they're using methods that vary tremendously and in ways that affect the quality and security of each piece and thus of the whole. Open-source software as a whole has an excellent reputation for security, and some projects prioritize earning that reputation. A recent study from Google's Project Zero found that the Linux operating system kernel fixed security holes in an average of 15 days during 2021, but another recent study from Sonatype found that 29 percent of major open-source projects contain known security vulnerabilities either in themselves or in the code that they require to function, their dependencies.

But this isn't just about defects in software. As we've heard, supply chain attacks, attacks on the way that developers and companies assemble and package and distribute their code, on top of many other dependencies, then—and then get to the end users, those kinds of attacks are on the rise. And several incidents over the last few years have led the open-source community to organize an array of different efforts to address the many underlying root causes that have combined to create the situation we're in.

The OpenSSF is home to many of these efforts, as I detail further in my written testimony. We are running programs and building solutions that bring greater trust and resiliency to the way that software flows through a supply chain through projects like sigstore which focus on—focuses on digital signatures for software artifacts, and SLSA, which tracks levels of process in the supply chain.

We are disturbing educational materials that teach the fundamentals of secure software development, something that few developers learn until later in their careers. We also are working heavily—we're working to identify the most critical software packages and measure them for security practices that can mitigate the risk of future major bugs, and if they aren't doing that work themselves, offer to help them do it. We are promoting these of third-

party audits to discover the architectural mistakes, the product misfeatures, and implementation oopsies that an attacker can wormhole through to get the goods and much more.

But we need help. The bad news is there is a lot of work to do and a lot of different kinds of work is needed. Now, I've just given a list. Now, the good news is we know what that work is, and we've got some proven tools and techniques that can scale up if the resources are made available. The great news is the returns on that investment are super scalable as those returns accrue to everybody using open-source software. You fix a hole once and everybody benefits no matter where you come from, how much you paid, all of that. If we focus those investments on the projects that are most critical, the most widely depended upon, and perhaps the ones that are least well-resourced, we can have a truly global impact.

In my written testimony I offered some advice on how the Federal Government can productively engage with the open-source software community on this topic. For this testimony I'd like to talk about some of the specific lines of effort that we are pursuing that would benefit from alignment with government efforts and opportunities. Briefly, they include expanding education on secure software development fundamentals. How do we get that everywhere, into every formal education around computer science, around every informal opportunity developers have to become better coders? How to help—we'd love to explore how to help industry develop better metrics and better benchmarks for measuring cyber risk in software itself. How do we know if we're succeeding at improving the data security and software?

We'd like to see a push for more supply chain integrity standards and tools from SBOMs (software bill of materials), which there has been great leadership on, to digital signatures and more, using procurement policy as a driver for change. We'd love to see a push for modern techniques such as memory-safe languages that can eliminate entire categories of software vulnerabilities. And we'd also love to see help with funding third-party code reviews for open-source projects. Think of them as audits but the good kind.

The good news is we are seeing a very proactive stance on this stance on this topic from the White House, the recent series of Executive orders and interagency efforts among the NSC (National Security Council), ONCD (Office of the National Cyber Director), OMB (Office of Management and Budget), OSTP, DOE (Department of Energy), HHS, and everybody has been really great to see. We're also really happy to see the interest from all of you on this topic. I believe the people and the systems that can drive systemic change are in place, but those efforts across both the public and private sector are currently resourced at a small fraction of what's needed to really solve the problem. There is so much to do. Thank you for your time.

[The prepared statement of Mr. Behlendorf follows:]

May 9th, 2022

The Honorable Eddie Bernice Johnson, Chairwoman
The Honorable Frank Lucas, Ranking Member
Committee on Science, Space, and Technology
2321 Rayburn House Office Building
Washington, DC 20515-6301

Dear Chairwoman Johnson, Congressman Lucas, and distinguished members of the Committee on Science, Space and Technology,

Thank you for your invitation to address you today, and the opportunity to share with you the work being done within the Open Source Security Foundation and the broader open source software community to raise the level of security and trustworthiness of open source software.

1. What are the consequences of insecure open-source software and what is industry as a whole, and the Open Source Security Foundation in particular, doing to tackle such Vulnerabilities?

Open source software ("OSS") has become an integral part of the technology landscape, as inseparable from the digital machinery of modern society as bridges and highways are from the physical equivalent. According to one report, typically 70% to 90% of a modern application "stack" consists of pre-existing OSS, from the operating system to the cloud container to the cryptography and networking functions, sometimes up to the very application running your enterprise or website. Thanks to copyright licenses that encourage no-charge re-use, remixing, and redistribution, OSS encourages even the most dogged of competitors to work together to address common challenges, saving money by avoiding duplication of effort, moving faster to innovate upon new ideas and adopt emerging standards.

However, this ubiquity and flexibility can come at a price. While OSS generally has an excellent reputation for security, the developer communities behind those works can vary significantly in their application of development practices and techniques that can reduce the risk of a defect in the code, or in responding quickly and safely when one is discovered by others. Often, developers trying to decide what OSS to use have difficulty determining which ones are more likely to be secure than others based on objective criteria. Enterprises often don't have a well-managed inventory of the software assets they use, with enough granular detail, to know when or if they're vulnerable to known defects, and when or how to upgrade. Even those enterprises who may be willing to invest in increasing the security of the OSS they use often don't know where to make those investments, nor their urgency relative to other priorities.

There are commercial solutions to some of these problems. There are vendors like Gitlab or Red Hat who sell support services for specific open source software, or even entire aggregate distributions of OSS. There are other vendors, like Snyk and Sonatype, who sell tools to help enterprises track their use of OSS and flash an alert when there is a new critical vulnerability in software running deep inside an enterprise's IT infrastructure.

However, fighting security issues at their upstream source - trying to catch them earlier in the development process, or even reduce the chances of their occurrence at all - remains a critical need. We are also seeing new kinds of attacks that focus less on vulnerabilities in code, and more on the supply chain itself - from rogue software that uses “typosquatting” on package names to insert itself unexpectedly into a developer’s dependency tree, to attacks on software build and distribution services, to developers turning their one-person projects into “protest-ware” with likely unintended consequences.

To address the urgent need for better security practices, tools, and techniques in the open source software ecosystem, a collection of organizations with deep investments into the OSS ecosystem came together in 2020 to form the Open Source Security Foundation, and chose to house that effort at the Linux Foundation. This public effort has grown to hundreds of active participants across dozens of different public initiatives housed under 7 working groups, with funding and partnership from over 75 different organizations, and reaching millions of OSS developers.

The OpenSSF’s seven working groups are:

1. **Best Practices for Open Source Developers**: This group works to provide open source developers with best practices recommendations, and easy ways to learn and apply them. Among other things, this group has developed courseware for teaching developers the fundamentals of secure software development, and implement the OpenSSF Best Practices Badge program.
2. **Securing Critical Projects**: This group exists to identify and help to allocate resources to secure the critical open source projects we all depend on. Among other things, this has led to a collaboration with Harvard Business School to develop a list of the most critical projects.
3. **Supply Chain Integrity**: This group is helping people understand and make decisions on the provenance of the code they maintain, produce and use. Among other things, this group has developed a specification and software called “SLSA”, for describing and tracking levels of confidence in a software supply chain.
4. **Securing Software Repositories**: This group provides a collaborative environment for aligning on the introduction of new tools and technologies to strengthen and secure software repositories, which are key points of leverage for security practices and the promotion to developers of more trustworthy software.
5. **Identifying Security Threats in Open Source Projects**: This group enables informed confidence in the security of OSS by collecting, curating, and communicating relevant metrics and metadata. For example, it is developing a database of all known security reviews of OSS.
6. **Security Tooling**: This group’s mission is to provide the best security tools for open source developers and make them universally accessible. Among other activities, this group has released code to better enable a security testing technique called “fuzzing” among open source projects.
7. **Vulnerability Disclosures**: This group is improving the overall security of the OSS ecosystem by helping advance vulnerability reporting and communication. For example, this group has produced a Guide to Coordinated Vulnerability Disclosure for OSS.

There are also a series of special projects under the OpenSSF worthy of special mention:

- **Project sigstore**: an easy-to-use toolkit and service for signing software artifacts, ensuring that the software you are holding is the same as what the developer intended, addressing a wide array of supply chain attacks.
- **The Alpha-Omega Project**: an effort to systematically search for new vulnerabilities in open source code, and work with critical open source projects to improve their vulnerability handling and other security practices.
- **The GNU Toolchain Initiative**: this effort supports the build ecosystems for perhaps the most critical set of developer libraries and compilers in the world, the GNU Toolchain, as a means to ensure its safety and integrity.

All the above efforts are public-facing and developed using the best practices of open source software communities. Funding from our corporate partners goes towards supporting the core staff and functions that enable this community, but all the substance comes from voluntary efforts. In some cases funds flow to assist with specific efforts - for example, recently the Alpha-Omega project decided to allocate funding towards the NodeJS community to augment its security team with a part-time paid employee and to fund fixes for security issues.

The Linux Foundation has also begun to adapt its "LFX" platform, a set of services designed to support the open source communities hosted by the Foundation, to incorporate security-related data such as vulnerability scans from Snyk and BluBracket, along with information from the OpenSSF Best Practices Badge program and the OpenSSF Security Scorecards initiative, to provide a unified view of the security risks in a particular collection of open source code, and what maintainers and contributors to those projects can do to improve those scores and reduce those risks. We expect to see more kinds of risk-related data coming into a unified view like this, helping developers and enterprises make better decisions about what open source components and frameworks to use, and how to reduce risk for those components they depend upon.

Guiding all of this is a deep conviction among the OpenSSF community that while there are many different ways in which security issues manifest themselves in the OSS ecosystem, every one of them is addressable, and that there are lots of opportunities for investment and collective action that will pay a return many times over in the form of lower risk of a future major vulnerability in a widely-used package, and lesser disruption if one is discovered.

Other efforts at the Linux Foundation include "Prossimo", an effort focused on moving core Internet-related services to "memory-safe" languages like Rust, Go, or Java, which would eliminate an entire category of vulnerabilities that other languages allow too easily. Another is the SPDX standard for Software Bill of Materials ("SBOMs"), addressing the needs identified by White House Executive Order 14028 in a vendor-neutral and open way.

This is by no means a comprehensive list of all such efforts in the OSS ecosystem to improve security. Every OSS foundation either has a security team in operation today or is scrambling to identify volunteers and funding to establish one. There is a greater emphasis today than I've seen in my 30 years of using and contributing to OSS

(since before it was called OSS) on the importance of such efforts. Clear metrics for progress are elusive since we lack clear metrics for evaluating software risk; in fact developing ways to measure and represent that risk is a key priority for OpenSSF. We will never see a time when open source software is free from security defects, but we are getting better at determining the tools and techniques required to more comprehensively address the risk of vulnerabilities in open source code. Scaling up those tools and techniques to address the tens of thousands of widely used OSS components and to get them more quickly updated remains a challenge.

2. How can the Federal government improve collaboration with industry to help secure open-source software?

I'll focus here on principles and methods for collaboration that will lead to more secure OSS, and then for question 3 on specific opportunities to collaborate on.

First, focus on resourcing long-term personal engagements with open source projects.

Over the last few years, we have seen a healthy degree of engagement by the Federal government with OSS projects and stakeholders on the topic of improving security. The push established by Executive Order 14028 for the adoption of SBOMs aligned nicely with the standardization and growing adoption of the SPDX standard by a number of OSS projects, but it was aided substantially by the involvement of personnel from NIST, CISA, and other agencies engaging directly with SPDX community members.

Often the real secret to a successful OSS effort is in the communities of different stakeholders that come together to create it - the software or specification is often just a useful byproduct. The Federal government, both through its massive use of open source code and the role that it traditionally performs in delivering and protecting critical infrastructure, should consider itself a stakeholder, and like other stakeholders prioritize engagement with upstream open source projects of all sizes. That engagement need not be so formal; most contributors to open source projects have no formal agreement covering that work aside from a grant of intellectual property in those contributions. But as they say, "history is made by those who show up." If the IT staff of a Federal agency (or of a contractor under a Federal contract) were authorized and directed to contribute to the security team of a critical open source project, or to addressing known or potential security issues in important code, or to participating in an OpenSSF working group or project, that would almost certainly lead to identifying and prioritizing work that would result in enhanced security in the Federal government's own use of open source code, and likely to upstream improvements that make OSS more secure for everyone else.

Second, engage in OSS development and security work as a form of global capacity building, and in doing so, in global stability and resilience. OSS development is inherently international and has been since its earliest days. Our adversaries and global competitors use the same OSS that we do, by and large. When our operating systems, cloud containers, networking stacks and applications are made to be more secure, there are fewer chances for rogue actors to cause disruption, and that can make it harder to de-escalate tensions or protect the safety of innocent parties. Government agencies in France, Taiwan, and more have begun to establish funded offices focused on the adoption, development, and promotion of OSS, in many ways echoing the Open Source

Program Offices being set up by companies like Home Depot and Walmart or intergovernmental agencies like the WHO. The State Department in recent years has funded the development of software like Tor to support the security needs of human rights workers and global activists. The Federal government could use its convening authority and statecraft to bring like-minded activities and investment together in a coordinated way more effectively than any of us in the private sector can.

Third, many of the ideas for improving the security of OSS involve establishing services - services for issuing keys to developers like Project sigstore does, or services for addressing the naming of software packages for SBOMs, or services for collecting security reviews, or providing a comprehensive view of the risk of open source packages. Wherever possible, the Federal government should avoid establishing such services themselves when suitable instances of such services are being built by the OSS community. Instead of owning or operating such services directly, the Federal Government should provide grants or other resources to operators of such services as any major stakeholder would. Along similar lines, should the Federal government fund activities like third party audits of an open source project, or fund fixes or improvements, it should ensure not only that such efforts don't duplicate work already being done, it should ensure that the results of that work are shared (with a minimum of delay) publicly and upstream so that everyone can benefit from that investment.

These three approaches to collaboration would have an outsized impact on any of the specific efforts that the Federal government could undertake.

3. Where should Congress or the Administration focus efforts to best support and secure the open-sourced software ecosystem as a whole?

The private sector and the Federal government have a common cause in seeing broad improvements in the security of OSS. I'm happy to share where I see the private sector starting to invest in enhanced OSS security, in the hopes that this may inspire similar actions from others.

1. **Education**. Very few software developers ever receive a structured education in security fundamentals, and often must learn the hard way about how their work can be attacked. The OpenSSF's Secure Software Fundamentals courses are well regarded and themselves licensed as open source software, which means educational institutions of all kinds could deliver the content. Enterprises could also start to require it of their own developers, especially those who touch or contribute to OSS. There must be other techniques for getting this content into more hands and certifications against it into more processes.
2. **Metrics and benchmarks**. There are plenty of efforts to determine what are suitably objective metrics for characterizing the risks of OSS packages. But running the cloud systems to perform that measurement across the top 100,000 or even 10,000 open source projects may cost more than what can be provided for free by a single company, or may be fragile if only provided by a single vendor. Collective efforts funded by major stakeholders are being planned-for now, and governments as a partner to that would not be turned away.
3. **Digital signatures.** There is a long history of U.S. Government standards for identity proofing, public key management, signature verification, and so on. These standards are very sophisticated, but in open

source circles, often simplicity and support are more important. This is pulling the open source ecosystem towards Project sigstore for the signing of software artifacts. We would encourage organizations of all sorts to look at sigstore and consider it for their OSS needs, even if it may not be suitable for all identity use cases.

4. **Research and development investments into memory-safe languages**. As detailed above, there are opportunities to eliminate whole categories of defects for critical infrastructure software by investing in alternatives written in memory-safe languages. This work is being done, but grants and investments can help accelerate that work.
5. **Fund third-party code reviews for top open source projects.** Most OSS projects, even the most critical ones, never receive the benefit of a formal review by a team of security experts trained to review code not only for small bugs that may lead to big compromises, but to look at architectural issues and even issues with the features offered by the software in the search for problems. Such audits vary tremendously in cost based on the complexity of the code, but an average for an average-sized code base would be $150K-250K. Covering the top 100 OSS projects with a review every other year, or even 200 every year, seems like a small price compared to the costs on US businesses to remedy or clean up after a breach caused by just one bug.
6. **Invest into better supply chain security support in key build systems, package managers, and distribution sites**. This is partly about seeing technologies like SBOMs, digital signatures, specifications like SLSA and others built into the most widely used dev tools so that they can be adopted and meaningfully used with a minimum of fuss. Any enterprise (including the Federal government) that has software certification processes based on the security attributes of software should consider how those tools could be enhanced with the above technologies, and automate many processes so that updates can be more frequent without sacrificing security.

These activities, if done at sufficient scale, could dramatically lower the risks of future disruptive events like we have seen. As a portfolio of different investments and activities they are mutually reinforcing, and none of them in isolation is likely to have much of a positive impact. Further econometrics research could help quantify the specific reduction of risk from each activity. But I believe that each represents a very cost-effective target for enhancing security in OSS no matter who is writing the check.

Thank you again for the opportunity to share these thoughts with you. I look forward to answering any questions you may have or providing you with further information.

Sincerely,

Brian Behlendorf
General Manager, Open Source Security Foundation
The Linux Foundation

548 Market Street PMB 57274 San Francisco, California 94104-5401
Phone/Fax: +1 415 723 9709 https://www.linuxfoundation.org/

Brian Behlendorf is the General Manager of the Open Source Security Foundation, a project hosted by the Linux Foundation. Brian has led the OpenSSF since September of 2021. Prior to that, from May of 2016, he was General Manager for Blockchain, Healthcare and Identity at the Linux Foundation, overseeing the Hyperledger and Linux Foundation Public Health initiatives. Prior to joining the Linux Foundation, Brian was Chief Technology Officer for the World Economic Forum, reporting to the founder Klaus Schwab and moving the organization into a more digital-first direction. In 2010 Brian worked as an advisor to the US Department of Health and Human Services on two open source initiatives, one called the Direct Project and the other the Connect Project, each focused on using OSS as a way to accelerate adoption of standards for the exchange of medical information. In 2009 he worked as an advisor in the Office of Science and Technology Policy, reporting to Deputy Chief Technology Officer Beth Noveck, with whom he worked through 2008 as an advisor to the Obama Campaign. From 1999 to 2007, Brian was the co-founder and CTO for CollabNet, a company focused on using open source software development tools and practices to accelerate software innovation. In 1995, Brian was a co-founder of the Apache httpd web server project and in 1998 the founding volunteer President of the Apache Software Foundation. Brian was the first Chief Engineer at Wired Magazine and a co-founder of the first Web design consultancy, Organic Online. Since 2003, Brian has served on the Board of Directors of the Mozilla Foundation. Since 2013, Brian has served on the Board of Directors of the Electronic Frontier Foundation.

Chairman FOSTER. Thank you. And next is Ms. Koran.

TESTIMONY OF MS. AMÉLIE ERIN KORAN, NON-RESIDENT SENIOR FELLOW, THE ATLANTIC COUNCIL

Ms. KORAN. Good morning, Chairman Foster and Chairwoman Stevens and Members of the Subcommittee. Thank you for the opportunity to testify before you today. While my written statement has the full details, I would like to summarize them for you here.

I'm Amélie Koran, Nonresident 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, Mr. Behlendorf, and Ms. Knausenberger.

The Cyber Statecraft Initiative strives to address strategic questions by combining systems analysis, policymaker engagement, and the operational experience of our interdisciplinary practitioner community. My views and perspectives expressed to you today come from the point of a contributor to open-source projects, a technician who's worked in operating and securing systems and critical infrastructure, and one lucky enough to experience all of this from both within the public and private sectors at various levels, as well as in different industries over the past quarter-century.

To quickly touch on the overarching questions posed to us by the Subcommittee, none of what you ask of us, the open-source community, or the agencies and programs you oversee could be easily answered, but there are ways to make it progressively better than its current state. First, realize that computer code is as much a part of our modern infrastructure that supports our country the same way more visibly tangible physical infrastructure like roads, bridges, dams, and utility plants are. However, there's a level of creativity and free expression involved that underpins how it's created, maintained, and used. It can be torn apart and excerpted, used in whole cloth as a complete package, or glued together with thousands of other pieces of code to achieve an end goal of building a system that thousands, millions, or billions of people use and rely on.

Addressing the security of this code, particularly of that which is developed to be shared and improved upon by primarily volunteer developers in an open and transparent fashion is why we're here to discuss the best ways for which this Committee, through its actions, can support its health and long-term viability.

While the Log4j vulnerability last year was the wake-up call that created this renewed focus, it was far from the first. Looking back a few years to Apache Struts, it was an open source package utilized by data broker Equifax, whose lack of following good maintenance and configuration best practices resulted in the loss of millions of individuals' personal data. This was an issue that could have been avoided by utilizing proper frameworks, guidance, and other nontechnical methodologies on how to properly manage and use and integrate that code into the digital environments.

Prior to that in 2014 the Heartbleed vulnerability found within an open-source software package used to secure the transmissions and communications across the internet was a project that lacked resources and sustainable government—governance to ensure something that did not actively introduce flaws into its critical code

base. Open-source software is one of the great often unacknowledged resources that allowed our modern society to advance and innovate like at no other time in its history through the adoption of openly available shared technology in our economy and daily lives.

It seems very easy to try to solve such issues by imposing regulations, more checklist-based compliance requirements, which tries to mask itself as security, or even turning away from using open-source software altogether. Just don't. This is an addressable problem through creative thinking, collaboration, and leveraging the best parts of industry, academia, and government, along with international partners to get this ecosystem to a point where it's healthy, sustainable, and, most of all, trustable.

Much like modern software development practices, everything should be iterative, and in my written testimony I am specific in saying the goal is to do the most good, which is not to try to solve it at first swipe but align things in such a way that they head in the right direction and then make our course corrects as we go along. Have government resources partner with organizations like the Open Source Security Foundation to leverage the interfaces into critical infrastructure through CISA (Cybersecurity and Infrastructure Security Agency) and NIST rather than have to reinvent the wheel and start a whole new line of engagements.

Leverage the grantmaking capabilities and appropriate oversight at various agencies to directly assist these types of foundations and efforts. It needs to do so agnostically through leveraging CISA's interfaces with sector coordination to identify and triage the most critical software packages that need these types of resources and efforts applied.

Use NIST and the NSF (National Science Foundation) to help develop, in conjunction with software stewardship foundations, such as the Linux and Apache Foundation and others, to help develop and refine standards and best practices. This should not end strictly at technical guidance but should also address how developers and project staff can best maintain and govern the entire lifecycle of their software projects. This work should also include consumer education on how to responsibly use, integrate, and operate code sourced from these efforts from the enterprise down to the individual consumers.

Exploit that same experience and trust from within these agencies and programs to build and assist with integrating validation frameworks for open-source projects deemed critical digital infrastructure even if it's just helping pair needs with existing resources. Often, it may just be awareness that they exist and they can have a measured positive affect.

Finally, be aware that none of this will be a quick fix. It requires a consistent, reliable effort from every group I've mentioned and from each angle to ensure success no matter how outwardly minor them may initially appear.

Thank you all, and I look forward to answering your questions.

[The prepared statement of Ms. Koran follows:]

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.

Amélie Koran is currently a Non-Resident Senior Fellow with The Atlantic Council, with a wide and varied background of nearly 30 years of professional experience in technology and leadership in the public and private sectors. During her career, she's supported work across various government agencies and programs including the US Department of the Interior, US Treasury Department, and US Department of Health and Human Services, Office of the Inspector General. In the private sector she's held various roles including those at The Walt Disney Company, Electronic Arts, Splunk, Constellation Energy, Mandiant, and Xerox.

She was detailed to the Executive Office of the President in 2014 to support the Federal CIO in reviewing cybersecurity legislation and was one of the original co-founders of the US Digital Service as part of the Presidential Management Council rotation program. Amélie is a graduate of Carnegie Mellon University, but was also a member of the Software Engineering Institute's CERT/CC in support of the Defense Cybercrime Center activities.

She is an avid volunteer and speaker within the security community, supporting various Security BSides events around the US and having spoken at DEF CON, ShmooCon, USENIX LISA, InfoSec World, AllDayDevOps and was profiled on Episode 91 of the Darknet Diaries podcast. Throughout her career she's been a staunch advocate for the importance of privacy, security, and sustainability of modern technology in a socially responsible fashion, including developing realistic career paths for current and incoming professionals to the field.

In her life outside technology and security, she's an avid motorcyclist, musician, skier, swimmer, and bicyclist. She lives with her spouse in a home to two wonderful cats and two rambunctious miniature schnauzer puppies. She writes on various technology and social topics on her personal blog "webjedi.net", as well as demystifies public policy topics and governmental operations on Twitter (@webjedi).

Chairman FOSTER. Thank you. And next is Dr. Lohn.

TESTIMONY OF DR. ANDREW LOHN, SENIOR FELLOW, CENTER FOR SECURITY AND EMERGING TECHNOLOGY, GEORGETOWN UNIVERSITY

Dr. LOHN. Thank you, Chairman Foster, Chairwoman Stevens, Ranking Member Obernolte, Ranking Member Feenstra, and Members of the Subcommittee. I'm Andrew Lohn, Senior Fellow in the CyberAI Project at the Center for Security and Emerging Technology at Georgetown University. It's an honor to be here.

During the next few minutes, I'd like to talk about the risks to the artificial intelligence supply chain. The AI community has been particularly open to sharing, for example, it cost half a million dollars in two and a half years to build the famous ImageNet data set, but the professor who built it released it to everyone. Then, Google and Facebook both released their powerful AI engines, and now thousands of the most powerful AI models are a quick download away. It's truly incredible, given that these models often range from thousands to millions of dollars to build, and that's just in the computing costs without even considering the expertise to design them.

These data sets, models, and AI programming resources form the building blocks of today's AI systems. In much the same way that few bakers today grow their own grain or raise their own hens, most AI developers simply combine ready-made components, then tweak them for their new applications. Sometimes that whole process only needs a few lines of code and surprisingly little expertise. This is the approach that allowed Google Translate to improve their performance with just 1/1000 of the code. They trimmed from 500,000 lines of code down to just 500.

That sharing has driven both scientific and economic progress, but it's also created an alluring target for attackers. For one, an attacker can subvert an AI system by altering the data. That could happen, for instance, by a nefarious online worker while they label the data sets or by an actor who sneaks into the victim's networks. Alternatively, if the attacker provides a fully trained model, then it can be very hard to find the manipulations. There's no good way to know if a downloaded model has a back door, and it turns out that those back doors can survive even after the system has been adapted for a new task. A poisoned computer vision system might mistake certain objects, or a poisoned language model might not detect terrorist messages or a disinformation campaign if they use the attacker's secret code words.

The programming resources for building AI systems are also vulnerable. Such systems can have thousands of contributors from around the globe writing millions of lines of code. Some of the code has been exploitable in the past, and some of it prioritizes speed or efficiency over security. For example, vision systems need images at a specific size, but the code to resize images allows attackers to swap one out for another.

And last, these resources are only as secure as the organizations or systems that provide them. Today, the vast majority are hosted in the United States or its allies, but China is making a push to create state-of-the-art resources and the network infrastructure to

provide them. If adversaries make the most capable models or if they simply host them for download, then developers in the United States would face an unwelcome choice between capability and security.

There are a few things that Congress can do now to help maximize the benefits of the sharing culture while limiting the security risks that come with it. One step is supporting efforts to provide trusted versions of these AI resources such as through NIST or a national AI research resource. Funding is also needed to do the basic hygiene, cleanup, and audits that are important for security but that attract few volunteers. Congress should consider requesting that organizations across the U.S. Government create a prioritized list of AI systems and resources used to build them. This list may be easier to create and maintain if those organizations are incentivized to collect a software bill of materials that list the components in the software that the government buys or builds.

And lastly, many of these AI systems are new and so are the attacks on them. The government would benefit from augmenting their teams of defensive hackers and security specialists with the AI expertise to help discover security holes in our most important systems. This would also allow them to think of new creative ways to subvert those systems before our adversaries do.

Thank you for the opportunity to testify today, and I look forward to your questions.

[The prepared statement of Dr. Lohn follows:]

Testimony before the House Science Subcommittee on Investigations and Oversight and Subcommittee on Research and Technology
Securing the Digital Commons: Open-Source Software Cybersecurity
May 11, 2022
Dr. Andrew Lohn

Chairman Foster, Chairwoman Stevens, Ranking Member Obernolte, Ranking Member Feenstra, and members of the Subcommittees, thank you for the opportunity to testify before you today. I am Andrew Lohn, Senior Fellow in the CyberAI Project of the Center for Security and Emerging Technology at Georgetown University. It is an honor to be here. During the next few minutes, I would like to discuss risks related to the artificial intelligence supply chain.

A Culture of Sharing

The AI community has been particularly open to sharing. For example, it cost $500,000 and two and a half years to build the famous ImageNet dataset, but the professor who built it released it to everyone. Then Google and Facebook both released their powerful AI engines. Now thousands of the most powerful AI models are a quick download away. It is truly incredible given that these models often range from thousands to millions of dollars to build – and that's in computing cost alone, without even considering the expertise to design them.

The AI Supply Chain

These datasets, models, and AI programming resources are the building blocks of today's AI systems. In the same way that few bakers today grow their own grain and raise their own hens, most AI developers simply combine ready-made components and tweak them for their new applications. Sometimes the whole process only needs a few lines of code and surprisingly little expertise. This approach allowed Google Translate to improve performance in 2016 while trimming from 500,000 lines of code down to just 500.

Sharing has driven both scientific and economic progress, but it has also created an alluring target for attackers.

Supply Chain Vulnerability

For one, an attacker can subvert an AI system by altering the data. That could happen, for instance, by a nefarious online worker while they label the datasets or by a hacker who sneaks into the victim's networks. Alternatively, if the attacker provides a fully trained model, then it can be very hard to find their manipulations.

There is no good way to know if a downloaded model has a backdoor, and it turns out that those backdoors can survive even after the system has been adapted for a new task. A poisoned computer vision system might mistakenly identify objects, or a poisoned language model might not detect terrorist messages or disinformation campaigns that use the attacker's secret code-words.

The programming resources for building AI systems are also vulnerable. Such systems can have thousands of contributors from around the globe writing millions of lines of code. Some of that code has been exploitable in the past. And some of it prioritizes speed or efficiency over security. For example, vision systems need images at a specific size, but the code to resize images allows attackers to swap out one image for another.

And lastly, these resources are only as secure as the organization or system that provides them. Today, the vast majority are hosted in the United States or its allies, but China is making a push to create state-of-the-art resources and the network infrastructure to provide them. If adversaries make the most capable models – or if they simply host them for download – then developers in the United States would face an unwelcome choice between capability and security.

Recommendations

There are a few things Congress can do now to help maximize the benefits of this sharing culture while limiting the security risks that come with it. One step is supporting efforts to provide trusted versions of these AI resources, such as through NIST or the National AI Research Resource. Funding is also needed to do the basic hygiene, cleanup, and audits that are important for security, but that attract few volunteers.

Congress should consider requesting that organizations across the U.S. government create a prioritized list of AI systems and the resources used to build them. This list may be easier to create and maintain if these organizations are incentivized to collect a software bill of materials that lists the components in the software that the government buys or builds.

And lastly, many of these AI systems are new, and so are the attacks on them. The government would benefit from augmenting their red and blue teams of defensive hackers and security specialists with AI expertise to help them discover security holes in our most important systems while also thinking of new, creative ways to subvert them before our adversaries do.

Thank you for the opportunity to testify today, and I look forward to your questions.

Andrew J. **Lohn**

PROFILE

Technical researcher at the core who has been able to step back and apply those skills and understanding to broad and complicated issues.

EDUCATION

Ph.D. - Electrical Engineering (2007-2012)
University of California Santa Cruz, CA USA

B.Eng. - Engineering Physics (2001-2006)
McMaster University, Hamilton, ON Canada

RESEARCH EXPERIENCE

CSET at Georgetown University – *Senior Fellow* - (2020-Present)
• Answer pressing policy problems at the three-way intersection of cybersecurity, artificial intelligence, and national security.
• Help develop future policy makers and advisors who have strong technical and methodological foundations along with broad policy perspectives.

RAND Corporation – *Information Scientist* - (2014-2020)
• Apply current methods to high-impact policy problems. Example methods: Reinforcement Learning, written equations, gaming, etc
• Lead teams of highly experienced researchers tackling complex problems. Example topics: AI risk, cyberwarfare, and drone delivery.
• Manage client relations with high-ranking executives in government

Pardee RAND Graduate School – *Professor of Public Policy* - (2018-2020)
• Design and teach course on offensive cybersecurity
• Mentor public policy graduate students, especially those with technical backgrounds or interests

Sandia National Laboratories - *Postdoctoral Researcher* - (2012-2014)
• Discovered and developed new device behavior then used it to design neural hardware for new computing architectures.
• Derived an equation describing the operation of next generation computing devices (RRAM) and used it to increase storage capacity per device by at least an order of magnitude.
• Our team went from TRL 0 (no working devices) to TRL 4 (wafer-scale, CMOS-compatible, device specs met) in one year, accelerating product timelines.

Hot Power, Inc. - *Chief Technology Officer* - (2011-2013)

• Led technology development and business planning for a nanotechnology-based energy company to convert heat to electricity.

NASA Ames Research Center - *Graduate Researcher* - (2009-2012)
• Built a nanotechnology lab from building permits to leading research facility.
• Attracted the interest of venture capitalists and government.

Hewlett-Packard Labs - *Visiting Researcher* - (2009-2012)
• Designed, simulated, and tested approaches to use light instead of electricity in computer wiring to alleviate a bottleneck in high performance computing

SELECTED AWARDS

Team Innovation Award – Project Air Force (2019).
Top 150 McMaster Engineering Alumni - 150th anniversary (2017).
RAND Spotlight Award (2015).
Sandia Certificate of Excellence (2013).
Newport Spectra Physics Research Excellence Award (2012).
APS Excellence in Graduate Research Award (2012).
Chancellor's Dissertation Fellowship (2011-2012).
National Graduate Student Award - American Vacuum Society - (2011).

COMMUNITY VOLUNTEER

IEEE Golden Reviewer Award - Electron Device Letters (2013, 2014, and 2016).
Outstanding Reviewer Award from Semiconductor Science and Technology (2017).

SELECTED TALKS

"Disinformation at Scale: Using GPT-3 Maliciously for Information Operations," Black Hat 2021.

"How Might AI Affect the Risk of Nuclear War?," Pentagon (2018) and Oxford University (2018).

"The future of urban air mobility," Uber Elevate 2018, Los Angeles, CA (2018).

"City-Scale Impacts of Drone Delivery," World Economic Forum - Future of Drones Steering Committee, San Francisco, CA (2017).

PUBLIC OPINION

Andrew J. Lohn, "What Chess Can Teach Us About the Future of AI and War," War On The Rocks, Jan 03, 2020
https://warontherocks.com/2020/01/what-chess-can-teach-us-about-the-future-of-ai-and-war/

Robert J. Lempert, Tim McDonald, **Andrew J. Lohn**, "A Better Way to Think About Scooters," Los Angeles Times Aug 28, 2018.
https://www.rand.org/blog/2018/08/a-better-way-to-think-about-scooters.html

Andrew J. Lohn, "What do Meltdown, Spectre and RyzenFall mean for the future of cybersecurity?" TechCrunch May 1, 2018.
https://techcrunch.com/2018/05/01/what-do-meltdown-spectre-and-ryzenfall-mean-for-the-future-of-cybersecurity/

Andrew J. Lohn, Edward Geist, "Will artificial intelligence undermine nuclear stability?" Bulletin of the Atomic Scientists Apr 30, 2018
https://thebulletin.org/will-artificial-intelligence-undermine-nuclear-stability11748

Andrew J. Lohn, Andrew Parasiliti, William Welser IV, "Should We Fear an AI Arms Race?" Defense One, Feb 08, 2016.
http://www.defenseone.com/ideas/2016/02/should-we-fear-ai-arms-race/125670/

Andrew J. Lohn, Andrew Parasiliti, William Welser IV, "How We Can Overcome the Risks of AI," TIME Magazine Oct 22, 2015
http://time.com/4080577/artificial- intelligence-risks/

WORK COVERED IN

BBC, Wall Street Journal, Forbes, POLITICO, CNBC, Wired, MIT Technology Review, Foreign Policy, Defense One, South China Morning Post, etc.

BOOK CHAPTERS

Andrew J. Lohn, Patrick R. Mickel, James B. Aimone, Matthew J. Marinella, "Memristors as Synapses in Artificial Neural Networks: Biomimicry Beyond Weight Change," in *Cybersecurity Systems for Human Cognition Augmentation*, Springer (2014).

RESEARCH REPORTS

Andrew J. Lohn, Wyatt Hoffman, "Securing AI: How Traditional Vulnerability Disclosure Must Adapt," Center for Security and Emerging Technology (2022).

Andrew J. Lohn, Micah Musser, "AI and Compute: How Much Longer Can Computing Power Drive Artificial Intelligence Progress," Center for Security and Emerging Technology (2022).

Ben Buchanan, **Andrew J. Lohn**, Micah Musser, Katerina Sedova "Truth, Lies, and Automation: How Language Models Could Change Disinformation," Center for Security and Emerging Technology (2021).

Andrew J. Lohn, "Poison in the Well: Securing the Shared Resources of Machine Learning," Center for Security and Emerging Technology (2021).

Andrew J. Lohn, "Hacking AI: A Primer for Policymakers on Machine Learning Cybersecurity," Center for Security and Emerging Technology (2020).

Andrew J. Lohn, Jair Aguirre, Mark Ashby, Benjamin Boudreaux, Jonathan Fujiwara, Gavin Hartnett, Daniel Ish, John Speed Meyers, Caolionn O'Connell, Li Ang Zhang, "Attacking Machine Learning in War," RR-4386-AF, (2020).

Forrest E Morgan, Benjamin Boudreaux, **Andrew J. Lohn**, Mark Ashby, Christian Curriden, Kelly Klima, Derek Grossman, "Military Applications of Artificial Intelligence: Ethical Concerns in an Uncertain World," RR-3139-AF (2020).

Li Ang Zhang, Jia Xu, Dara Gold, Jeff Hagen, Ajay K. Kochhar, **Andrew J. Lohn**, Osonda A. Osoba, "Air Dominance Through Machine Learning – A Preliminary Exploration of AI-Assisted Mission Planning," RR-4311-RC (2020).

Zachary Haldeman, Jair Aguirre, Jonathan Fujiwara, **Andrew Lohn**, Igor Mikolic- Torreira, "Effects Estimation for Cyberspace Operations," RR-3090-OSD, (2019).

Andrew J. Lohn, Quentin E. Hodson, "Quick Look: State Election Security Needs: An Analysis of the 2018 Help America Vote Act State Plans," PR-4347-DHS (2019).

Andrew Lohn, Akhil Shah, Jair Aguirre, Igor Mikolic-Torreira, "Uncertainty Analysis for Offensive Cyberspace Operations Effects Estimations," PR-3716-AF/1, (2019).

Andrew Lohn, Joshua Baron, Akhil Shah, Lillian Ablon, Irina Danescu, Lara Schmidt, "Uncertainty Analysis for Offensive Cyberspace Operations Effects Estimations," PR-3716-AF/1, (2019).

Andrew Lohn, Akhil Shah, Jair Aguirre, Dara Gold, "Uncertainty Analysis for Offensive Cyberspace Operations Effects Estimations," RR-2381-AF, (2019).

Edward Geist, **Andrew J. Lohn**, "Will Artificial Intelligence Increase the Risk of Nuclear War," PE-296-RC (2018).

Andrew J. Lohn, et al., "Providing Cyber Mission Assurance for Weapon Systems: An F-16 Case Study," RR-2838-AF (2019).

Caolionn O'Connell, et al., "Assessing Cybersecurity Risk to the Civil Engineering Infrastructure: A Methodology for Implementation at Air Force Bases," RR-2354-AF (2018).

Caolionn O'Connell, et al., "Cybersecurity of USAF Civil Engineering Control Systems: Buckley Air Force Base Case Study," (2017).

Andrew J. Lohn, Lara Schmidt, Caolionn O'Connell, Joshua Baron, "Results of a Wargame to Improve the Utility and Efficiency of Operational Test for Cyber Weapons," RR-1897-OSD, (2017).

Andrew J. Lohn, "The City-Scale Impacts of Drone Delivery," RR-1718-RAND, (2017).

Bryan W. Hallmark, et al. “Using CTC-Based Metrics to Support Policy and Program Decisions,” (2016).

Lara Schmidt, et. al. “Effects Estimation for Offensive Cyber: Is it Time for a Cyber JMEM?” (2016).

Yool Kim, et. al. “Assessing the Risks of Commonality Between Ground Based Strategic Deterrent and Submarine-Launched Ballistic Missile Systems.” (2016).

Jennie W. Wenger, et. al. “The Value of Experience in the Enlisted Force” (2016).

Conrad D. James, et. al. “A comprehensive approach to decipher biological computation to achieve next generation high-performance exascale computing” SAND2013-7915 (2013).

PATENTS

Andrew J. Lohn, Patrick R. Mickel, “Multilevel Resistive Information Storage and Retrieval,” US Patent No. 9,412,446 (2016).

P.R. Mickel, C.D. James, **Andrew J. Lohn**, M.J. Marinella, “Methods for resistive switching of memristors," US Patent No 9,336,870 (2016).

James E. Stevens, Matthew Marinella, **Andrew J. Lohn** Aluminum, "Memristor Using a Transition Metal Nitride Insulator," US Patent No. 8,872,246 (2014).

Nobuhiko P. Kobayashi and **Andrew J. Lohn**. Nanowire Composite for Thermoelectrics. WO 2,013,043,926 (Sept 20, 2012).

Patrick R. Mickel, Conrad D. James, Matthew J. Marinella, **Andrew J. Lohn**, "Method for Measuring and Modifying Memristor Switching Characteristics," Provisional App. No. 61/894,816 (Oct. 23, 2013).

James E. Stevens, **Andrew J. Lohn**, Patrick R. Mickel, Matthew J. Marinella, "Systems and methods to maintain optimum stoichiometry for reactively sputtered films," Provisional App. No. 61/971,301 (Jun. 21, 2013).

James B. Aimone, **Andrew J. Lohn**, Patrick R. Mickel, Erik P. Debenedictis, "Memristor circuit implementation of neurogenesis in neural networks," TA SD# 12953

Andrew J. Lohn, Patrick R. Mickel, Matthew J. Marinella, "Electrode Design for High Retention Resistive Switching," TA SD# 12945 (Nov. 25, 2013).

PUBLICATION

56) Andrew J. Lohn, “Downscaling Attack and Defense: Turning What You See Back Into What You Get,” arXiv 2010.02456 (2020).

55) Andrew J. Lohn, "Estimating the Brittleness of AI: Safety Integrity Levels and the Need for Testing Out-Of-Distribution Performance," arXiv 2009.00802 (2020).

54) M. Brundage et al, "Toward Trustworthy AI development: mechanisms for supporting verifiable claims," arXiv arXiv 2004.07213 (2020).

53) Gavin S. Hartnett, **Andrew J. Lohn**, Alexander P. Sedlack, "Adversarial Examples for Cost-Sensitive Classifiers," Proceedings of the 33rd Conference on Neural Information Processing Systems - NeurIPS (2019).

52) Daniel Ish, **Andrew Lohn**, Christian Curriden, "A Quantitative History of A.I. Research in the United States and China," in review, WR-1318-AF (2019).

51) Andrew J. Lohn, "Defense in Depth: The Basics of Blockade and Delay," arXiv:1910.00111, in review, (2019).

50) Andrew J. Lohn, "Timelines for In-Code Discovery of Zero-Day Vulnerabilities and Supply-Chain Attacks," arXiv:1808.10062 (2018).

49) B.J. Choi, A.C. Torrezan, J.P. Strachan, P.G. Kotula, **Andrew J. Lohn**, Matthew J. Marinella, Z. Li, R.S. Williams, J.J. Yang "High-Speed and Low-Energy Nitride Memristors," Advanced Functional Materials (2016).

48) Patrick R. Mickel, David Hughart, **Andrew J. Lohn**, Xujiao Gao, Dennis Mamaluy, Matthew J. Marinella, "Power signatures of electric field and thermal switching regimes in memristive SET transitions," Journal of Physics D: Applied Physics, **49**, 245103 (2016).

47) Patrick R. Mickel, **Andrew J. Lohn**, Dennis Mamaluy, Matthew J. Marinella, "Power signatures and vacancy profile control in nanoscale memristive filaments," Applied Physics Letters **107**, 033507 (2015).

46) D.R. Hughart, **A.J. Lohn**, P.R. Mickel, E. Bielejec, G. Vizkelethy, B.L. Doyle, S.L. Wolfley, P.E. Dodd, M.R. Shaneyfelt, M.L. McLain, M.J. Marinella, "Mapping of Radiation-Induced Resistance Changes and Multiple Conduction Channels in TaOx Memristors," IEEE Transactions on Nuclear Science, **61**, 2965-2971 (2014).

45) Andrew J. Lohn, Patrick R. Mickel, Matthew J. Marinella, "Modeling of filamentary resistive memory by concentric cylinders with variable conductivity," Applied Physics Letters **105**, 183511 (2014).

44) Matthew J. Marinella, Patrick R. Mickel, **Andrew J. Lohn**, David R. Hughart, Robert Bondi, Denis Mamluy, Harold P. Hjalmarson, James E. Stevens, Seth Decker, Roger T. Apodaca, Brian Evans, James Bradley Aimone, Fred Rothganger, Conrad D. James, Erik P. Debenedictis, "Development, Characterization, and Modeling of a TaOx ReRAM for a Neuromorphic Accelerator," ECS Transactions **64**, 37-42 (2014).

43) Patrick R. Mickel, **Andrew J. Lohn**, Matthew J. Marinella, "Detection and characterization of multi-filament evolution during resistive switching," Applied Physics Letters **105**, 053503 (2014).

42) D.R. Hughart, **A.J. Lohn**, P.R. Mickel, P.E. Dodd, M.R. Shaneyfelt, A.I. Silva, E. Bielejec, G. Vizkelethy, B.L. Doyle, M.T. Marshall, M.L. McLain, M.J. Marinella, S.M. Dalton, "Radiation-induced resistance changes in TaOx and TiO2 memristors," IEEE Aerospace Conference 1-11 (2014).

41) Michael T. Brumbach, Patrick R. Mickel, **Andrew J. Lohn**, Alex J. Mirabal, Michael A. Kalan, James E. Stevens, M.J. Marinella, "Evaluating tantalum oxide stoichiometry and oxidation states for optimal memristor performance," Journal of Vacuum Science and Technology A, **32**, 051403 (2014)..

40) Andrew J. Lohn, Patrick R. Mickel, Matthew J. Marinella, "Analytical estimations for thermal crosstalk, retention, and scaling limits in filamentary resistive memory," Journal of Applied Physics **115**, 234507 (2014).

39) Andrew J. Lohn, Patrick R Mickel, Matthew J. Marinella, "Mechanism of electrical shorting failure mode in resistive switching," Journal of Applied Physics **116**, 034506 (2014).

38) Andrew J. Lohn, Patrick R. Mickel, Conrad D. James, Matthew J. Marinella, "Degenerate Resistive Switching and Ultrahigh Density Storage in Resistive Memory," Applied Physics Letters **105**, 103501 (2014).

37) Patrick R Mickel, **Andrew J. Lohn**, Matthew J. Marinella, "Memristive Switching: Physical Mechanisms and Applications," Modern Physics Letters B **28**, 1430003 (2014).

36) Andrew J. Lohn, Patrick R. Mickel, et al, "Isothermal Switching and Detailed Filament Characterization in Resistive Switches", Advanced Materials **26**, 4486-4490 (2014).

35) David Hughart, **Andrew J. Lohn**, Patrick R. Mickel, Scott P. Dalton, Paul E. Dodd, Marty R. Shaneyfelt, Ed Bielejec, George Vizkelethy, M.T. Marshal, Matthew J. Marinella, "A Comparison of the Radiation Response of TaOx and TiO2 Memristors," IEEE Transactions on Nuclear Science **60**, 4512-4519 (2014).

34) Andrew J. Lohn, Barney L. Doyle, Patrick R. Mickel, Matthew J. Marinella, "Rutherford Forward Scattering and Elastic Recoil Detection," Nuclear Instruments and Methods B, **332**, 99-102 (2014).

33) James E. Stevens, **Andrew J. Lohn**, Seth A. Decker, Patrick R. Mickel, Matthew J. Marinella, "Reactive sputtering of substoichiometric Ta2Ox for resistive memory applications," Journal of Vacuum Science and Technology A, **32**, 021501 (2013).

32) Matthew J. Marinella, James E. Stevens, Patrick R. Mickel, David R. Hughart, **Andrew J. Lohn**, "A CMOS Compatible, Forming Free TaOx ReRAM," ECS Transactions **58**, 59-65 (2013).

31) Andrew J. Lohn, Patrick R. Mickel, Matthew J. Marinella, "Dynamics of Percolative Breakdown Mechanism in Tantalum Oxide Resistive Switching," Applied Physics Letters **103**, 173503 (2013).

30) Andrew J. Lohn, James E. Stevens, Patrick R. Mickel, Matthew J. Marinella, "Optimizing TaOx memristor performance and consistency within the reactive sputtering "forbidden region"," Applied Physics Letters **103** 063502 (2013).

29) Patrick R. Mickel, **Andrew J. Lohn**, Byung Joon Choi, J.Joshua Yang, Min-Xian Zhang, Matthew J. Marinella, Conrad D. James, R. Stanley Williams, "A physical model of switching dynamics in tantalum oxide memristive devices," Applied Physics Letters **102** 223502 (2013).

28) Andrew J. Lohn, Robert D. Cormia, David M. Fryauf, Junce Zhang, Kate J. Norris, Nobuhiko P. Kobayashi, "Morphological Effect of Doping Environment on Silicon Nanowires Grown by Plasma-Assisted Chemical Vapor Deposition", Japanese Journal of Applied Physics **51** p.11 (2012).

27) Andrew J. Lohn, Noel Dawson, Robert Cormia, David Fryauf, Junce Zhang, Kate J. Norris, Nobuhiko P. Kobayashi, "Study on indium phosphide nanowires grown by metal organic chemical vapor deposition and coated with aluminum oxides deposited by atomic layer deposition", SPIE NanoScience + Engineering, 84670U-6 (2012).

26) Jin-Woo Han, **Andrew J. Lohn**, Meyya Meyyappan, Nobuhiko P. Kobayashi, "Contact metal effects in indium phosphide nanowire transistor", SPIE Nanoscience + Engineering 84670Z-6 (2012).

25) Kate J. Norris, Junce Zhang, David M. Fryauf, Allison Rugar, Amanda Flores, Timothy J. Longson, **Andrew J. Lohn**, Nobuhiko P. Kobayashi, "Indium phosphide nanowire network: growth and characterization for thermoelectric conversion", SPIE NanoScience + Engineering 84670E-8 (2012).

24) Kate J. Norris, **Andrew J. Lohn**, Elane Coleman, Gary S. Tompa, Nobuhiko P. Kobayashi, "Modeling and Characterization of Silicon Nanowire Networks for Thermoelectric Conversion", MRS Proceedings 1456 p.1 (2012).

23) Kate J. Norris, Vernon Wong, Takehiro Onishi, **Andrew J. Lohn**, Elane Coleman, Gary S. Tompa, Nobuhiko P. Kobayashi, "Reflection Absorption Infrared Spectroscopy Analysis of the Evolution of ErSb on InSb", Surface Science (2012).

22) Andrew J. Lohn, Kate Norris, Robert D. Cormia, Elane Coleman, Gary S. Tompa, Nobuhiko P. Kobayashi, "Effect of Doping on Nanowire Morphology During Plasma-Assisted Chemical Vapor Deposition", MRS Proceedings 1439 p.1 (2012).

21) **Andrew J. Lohn**, Nobuhiko P. Kobayashi, "AC Surface Photovoltage of Indium Phosphide Nanowire Networks", Applied Physics A, **107** pp 647-651 (2012).

20) Kate J. Norris, **Andrew J. Lohn**, Elane Coleman, Vernon Wong, Ali Shakouri, Gary S. Tompa, Nobuhiko P. Kobayashi, "MOCVD growth of erbium monoantimonide thin film and nanocomposites for thermoelectrics" – Journal of Electronic Materials p.1 (2012).

19) **Andrew J. Lohn**, Elane Coleman, Gary S. Tompa, Nobuhiko P. Kobayashi, "Assessment on thermoelectric power factor in silicon semiconductor nanowire networks", Physica Status Solidi A **209** pp. 171-175 2012.

18) Takehiro Onishi, **Andrew J. Lohn**, Elane Coleman, Gary S. Tompa, Nobuhiko P. Kobayashi, "Reflection Absorption Infrared Spectroscopy Study on the Spontaneous Formation of Erbium Monoantimonide Nanoparticles on Indium Antimonide Surfaces", MRS Proceedings 1351 p.1 (2011).

17) **Andrew J. Lohn**, Timothy J. Longson, Nobuhiko P. Kobayashi, "Indium phosphide nanowires integrated directly on carbon fiber", Proc. SPIE 81060X (2011).

16) Takehiro Onishi, Kate J. Norris, **Andrew J. Lohn**, Vernon Wong, Nitish Padgaonkar, Elane Coleman, Gary S. Tompa, Nobuhiko P. Kobayashi, "Nanocomposites for thermoelectric power generation: rare-earth metal monoantimonide nanostructures embedded in InGaSb and InSbAs ternary alloys", Proc. SPIE 81060Q (2011).

15) Jin-Woo Han, **Andrew J. Lohn**, Nobuhiko P. Kobayashi, Meyya Meyyappan, "Copper oxide thin film and nanowire for e-textile applications", Proc. SPIE 810608 (2011).

14) Toshishige Yamada, Hidenori Yamada, **Andrew J. Lohn**, Nobuhiko P. Kobayashi, "Transport in fused indium phosphide nanowire device in dark and under illumination: Coulomb staircase scenario", Proc. SPIE 81060I (2011).

13) Jin-Woo Han, **Andrew J. Lohn**, Nobuhiko P. Kobayashi, Meyya Meyyappan, "Evolutional Transformation of Copper Oxide Nanowires to Copper Nanowires by a Reduction Technique", Materials Express **1** pp. 176-180 (2011).

12) **Andrew J. Lohn**, Jin-Woo Han, Nobuhiko P. Kobayashi, "Surface Photovoltage Study of Indium Phosphide Nanowire Networks", Accepted for Publication in Proceedings of the Materials Research Society (2011).

11) Toshishige Yamada, Hidenori Yamada, **Andrew J. Lohn**, Nobuhiko P. Kobayashi, "Room-Temperature Coulomb Staircase in Semiconducting InP Nanowires Modulated with Light Illumination", Nanotechnology, **22** 055201 (2010).

10) **Andrew J. Lohn**, Xuema Li, Nobuhiko P. Kobayashi, "Epitaxial growth of ensembles of indium phosphide nanowires on various non-single crystal substrate using an amorphous template layer", Journal of Crystal Growth, **315** pp. 157-159 (2010).

9) **Andrew J. Lohn**, Milo Holt, Noel Dawson, Nobuhiko P. Kobayashi, "Ensemble Effects on the Optical Properties of Indium Phosphide Nanowires at Various Temperatures", Proceedings of the Materials Research Society, 1258 P04-14 (2010).

8) **Andrew J. Lohn**, Takehiro Onishi, Nobuhiko P. Kobayashi, "Optical properties of indium phosphide nanowire ensembles at various temperatures", Nanotechnology, **21** pp. 355702 (2010).

7) **Andrew J. Lohn**, Nobuhiko P. Kobayashi, "Effect of Substrate Crystallinity on Gr owth and Optical Properties of InP Nanowires", Proceedings of the IEEE Nanotechnology, Materials and Devices Conference pp. 169 (2010).

6) Hidenori Yamada, Toshishige Yamada, **Andrew J. Lohn**, Nobuhiko P. Kobayashi, "Reversible Suppression of Coulomb Staircase in InP Nanowires with Light Illumination", Proceedings of the IEEE Nanotechnology Materials and Devices Conference pp. 305 (2010).

5) **Andrew J. Lohn**, Milo Holt, Noel Dawson, Nobuhiko P. Kobayashi, "Temperature dependent optical properties of InP nanowire ensembles", Proceedings of the SPIE, pp. 767920 (2010).

4) Hidenori Yamada, Toshishige Yamada, **Andrew J. Lohn**, Nobuhiko P. Kobayashi, "Coulomb staircase in fused InP nanowires under light illumination", Proceedings of the SPIE **4** pp. 7768-10 (2010).

3) **Andrew J. Lohn**, Takehiro Onishi, Nobuhiko P. Kobayashi, "Characterization of nanowires grown on non-single crystal platforms", Proceedings of the SPIE pp. 73180C-1 (2009).

2) Takehiro Onishi, **Andrew J. Lohn**, Nobuhiko P. Kobayashi, "Optical Properties and Carrier Dynamics of Ensembles of InP Nanowires Grown on Non-Single Crystal Platforms", Proceedings of the Materials Research Society 1178-AA01-04 (2009).

1) Nobuhiko P. Kobayashi, Sagi, Mathai, Xuema Li, V.J. Logeeswaran, M. Saif Islam, **Andrew J. Lohn**, Takehiro Onishi, Joseph Straznicky, Shih-Yuan Wang, R. Stanley Williams, "Ensembles of indium phosphide nanowires: physical properties and functional devices integrated on non-single crystal platforms", Applied Physics A, **95** pp. 1005-1013 (2009).

Chairman FOSTER. Thank you. And at this point we will begin our first round of questions. The Chair now recognizes himself for five minutes.

Now, digital identity management and digital ID and digital signatures are critical to prevent security breaches. This set of problems occurs really at two different levels. The first thing, during co-development, the co-developers need to be able to prove they are who they say they are for many obvious reasons. And second, during code execution, the so-called zero trust architecture that I think a lot of very secure environments are moving toward, needs to continuously check the authorizing credentials of code components as they execute.

And so, Mr. Behlendorf, in your testimony you talk about the need to develop simple digital signature guidance and encourage the open-source community to adopt it. Could you elaborate a little bit on the challenges there and how the Federal Government and governments around the world, at least the free democracies of the world, should approach this?

Mr. BEHLENDORF. Yes, well, we've had quite a few different systems for—to be able to sign emails, being able to sign digital artifacts of all sort going back to PGP (Pretty Good Privacy) in the—which started in the late 1980's, early 1990's. In fact PGP and derivatives of it do form an important part of the last mile when it comes to validating the packages you get from repositories are what they say they are. Many development teams sign them as well. And yet for its length, PGP and GPG (GNU Privacy Guard) have not really reached the level of ubiquity across the software supply chains that have really been needed.

So one of the projects of the OpenSSF is called project sigstore, and this brings a much simpler approach to being able to get keys to sign an artifact and push it through the supply chain, ensuring that humans are actually the ones doing the signing so that you get that check that's not—there are parts that are automated appropriately and other parts that do require human oversight to make that actually meaningful.

This is something that's now picked up quite a bit of adoption. The major cloud container system called Kubernetes has now adopted it ubiquitously. And it also interfaces nicely with other ID systems out there.

I think the appropriate role for governments in general is to be supportive of these efforts that are emerging from the open-source community. As always with digital identity, we do have this question of where—what are the roots of the public key infrastructure from which this comes? And I would really encourage the government to look at the example set by ICANN (Internet Corporation for Assigned Names and Numbers) and its administration of the domain name system or the CA/Browser Forum in its administration of the root certificates in web browsers and see that there are approaches to managing trust at scale. And these technologies will likely converge on that same kind of distributed systems for being able to manage the roots of that trust.

Lots of other efforts going on, and I'd be happy to share more in the fullness of time. Thank you for the question.

Chairman FOSTER. Yes. And I think one of the toughest issues there is the—sort of the root of identity, which I fear is always going to be an essential government function. To make, you know, essentially a list of legally traceable human beings for high-security applications, and that's going to be something that the free democracies of the world are going to have to work together to make sure that that kind of identity system, you know, basically interoperates properly.

Now, in terms of the other—the zero trust thing that I mentioned, you know, last year, the Administration's Executive Order 14028 specifically called on Federal agencies to develop a plan to implement zero trust architecture. The idea of zero trust is that a network will continuously check on a user to make sure that they are who they say they are and that the software they're running carries that down through the system. And so then the adversary won't be able to just run amok once they gain access inside the system.

So, Ms. Knausenberger, Platform One has developed a zero trust software called Cloud Native Access Point, or CNAP. Can you talk about the lessons the Air Force has learned in implementing the system and how open-source developers could utilize it?

Ms. KNAUSENBERGER. Absolutely, and thank you for the question. First, I will share that zero trust has been a huge priority for the Department of the Air Force, as well as the Department of Defense coming into the incoming budget cycle with a lot of really valuable pilots ongoing now. The Cloud Native Access Point has been one of the most mature and successful pilots that we have run in that we've proven that we don't have to use the typical gated "castle" approach, which, as you noted, once you get an adversary into an older model, they can just move freely.

We've shown that we can drive better performance, that we can drive greater speed, and that we can do so more securely. We have extensively red-teamed the solution and continue to make improvements. We've also shown that we can move more quickly as we make those improvements. We do have a lot to do, I'd say, at the enterprise-level on identity, but the Platform One team and our CNAP approach are very much leading the way to show us the realm of the possible.

Chairman FOSTER. Thank you. And my time is up and I now recognize Mr. Obernolte for five minutes.

Mr. OBERNOLTE. Thank you, Mr. Chair. Thank you to the witnesses. This has been a fascinating hearing so far.

Ms. Knausenberger, let me continue that line of questioning that Chairman Foster started on the—on Platform One. Now, obviously, that's been the Department of Defense's answer to how to secure software that's built on open source. In particular in your testimony you mentioned Iron Bank, which is the Air Force's way of protecting against supply chain vulnerabilities where a malicious actor might introduce intentionally a vulnerability in the hopes that it would be incorporated in an end product in a sensitive area. Can you talk a little bit about how that might be replicated for uses outside of the Department of Defense?

Ms. KNAUSENBERGER. Certainly. So most private companies and most development teams do have a process for checking code that

they bring in. They leverage their CI/CD pipeline, which typically would include things like static and dynamic code analysis, dependency checking, looking for secrets, fuzz testing. You know, there's a typical set of tools that different entities would go through to ensure the security of that code.

The way that we've done it is instead of pushing it out to independent development teams to do that, however they have determined their best practices, to help them centrally by going through and doing things like ensuring the code came from where we thought it came from, scanning containers, scanning code, consistently looking at the hashing, automatically updating the containers, and providing that as a common repo where folks can point to that container in the repo and leverage that code.

We have had Fortune 500 companies also use that—our repo, our containers. We have had one commercial bank use our containers. It is available to the public to use, and we have had some great contribution. Over 300 commercial entities have contributed back to ensuring that Iron Bank is continuously improved as well.

Mr. OBERNOLTE. Great, thank you. That sounds like great work. Mr. Behlendorf, if I could throw a question your way, in your testimony you mentioned memory-safe languages as being a promising area for reducing security risks. And I find that a very interesting topic because the research on the number of security problems that are caused by memory-safety violations is just stunning. There was a recent study that said over 60 percent of all vulnerabilities in Apple software is as a result of memory problems. It's—Microsoft estimates it's over 70 percent for Microsoft software, and Google says it's over 90 percent for android vulnerabilities. So, you know, I think this is a really interesting conversation to be having.

And obviously using memory-safe language would prohibit the kind of behavior that is causing these problems, you know, which would—like the Heartbleed bug that the Chairman mentioned in SSL, that was an out-of-bounds read that would have been prohibited by memory-safe language. WannaCry was an out-of-bounds right. So, you know, this is a really—this would solve a lot of problems.

But I'll tell you this, as a programmer, when I was writing software, I hated memory-safe languages. And the reason is performance because if you got something every time you want to read from memory or write for memory, you've got this, you know, cyber overlord that's determining whether or not you're allowed to do it. That takes performance away from your software. And in an era where we're talking about AI applications that are very performance-intensive, you know, how do you balance those two priorities where we're giving up, you know, performance by a factor of probably two or three times to solve problems that occur, you know, once in a trillion times? So, you know, how do you balance that—those two priorities when you're talking about memory safety?

Mr. BEHLENDORF. It's a great question, and this was one place where Moore's Law continues to be on our side, right, because——

Mr. OBERNOLTE. Well, it's——

Mr. BEHLENDORF. —[inaudible].

Mr. OBERNOLTE. —[inaudible].

Mr. BEHLENDORF. The cost of being able to perform a certain number of operations per second continues to drop. The cost of storage, all that continue their projections, right? So—but we also found in the last couple of years real advances in languages like Rust and Go that have allowed us to build memory-safe functions in—well, software in memory-safe languages that approach, in some cases even exceed, the performance of C code and some really low-level functions. So there's been really exciting advances in the last few years, and I think the time is ripe to really consider looking at a lot of fundamental libraries and parts of the internet architecture such as the software that runs the domain system as opportunities to, again, eliminate entire categories of software vulnerabilities.

Mr. OBERNOLTE. Sure. So how do you do that? Do you evangelize memory-safe languages through educational outreach to programmers that say that some of these——

Mr. BEHLENDORF. You also——

Mr. OBERNOLTE. Yes?

Mr. BEHLENDORF. Yes, sir, I'm sorry. You also try to resource the teams to go and build that software directly, again, very highly leveraged because it can be shared with everybody very quickly. But it doesn't take a whole lot, a few developers at the peak of their level if you want folks to evangelize and help build that. But the number of people who run domain name servers out there is actually—you could probably count them on three or four hands to cover the majority of the internet. And so there are some—a couple of very strategic places were you could have a tremendous impact.

Mr. OBERNOLTE. Right. Thank you. Well, I see I'm out of time. I yield back, Mr. Chair.

Chairman FOSTER. I now recognize Representative Stevens for five minutes.

Ms. STEVENS. Great, thank you.

As I mentioned in my opening statement, the *America COMPETES Act* directs NIST to address or assess security risks in open-source software and create guidance to help organizations maintaining open-source code. Ms. Koran, we've heard from obviously many stakeholders that NIST's cybersecurity guidance is often hard to adopt, especially by smaller entities. And this seems even more challenging in the open-source context. Where are many software packages are managed by—you know, they're managed by smaller teams, so what steps can NIST take to make it easier or just to make it as easy as possible for the open-source community to adopt NIST cybersecurity guidance?

Ms. KORAN. Well, one of the problems that I've noticed, you know, in my time in both industry and the public sector is the digestibility of it. And it's very easy for government to kind of spend all the resources that they want to kind of, you know, get the tool—tooling and stuff in place. One of the biggest challenges there is to adopt those standards and frameworks into tools that are consumable by smaller groups such as developers or smaller and medium-size businesses that don't have those types of resources. So potentially, you know, pairing with some of the places like Open Source Security Foundation and whatnot, that is a—you know, using the same open-source type of model to provide those, you

know, audit capabilities in those packages that are kind of—I wouldn't say downscale but more or less a community addition in a way to allow them to, you know, consume and then apply those standards and guidances.

The other issue, too, with NIST is it's also very academic. It needs to be written in a plain language that someone can basically—a more simple guidebook to address, you know, how to apply that to their own current problems. A lot of the stuff that, as written, is not situational to, you know, particular—you know, a small business, you know, is putting someone in their shoes as you're writing guidance. Those are some steps that NIST can take as to kind of think less of the—more like I'm an academic writer here to basically create these controls but apply them more like, hey, I'm in a certain situation and here's kind of a step through to work through those processes.

Ms. STEVENS. Great. So I also believe that education and training will be a key part of securing the open-source ecosystem. Many developers aren't focused on security and may not know what resources and tools exist to ensure that they are developing secure software. Ms. Koran, could you elaborate on how we can leverage the National Initiative for Cybersecurity Education at NIST to help make open-source developers more security-minded?

Ms. KORAN. One of the things is marketing. Awareness is one of the biggest challenges. The security community is I wouldn't say necessarily insular. We exchange ideas at security conferences like Defcon or BSides and so forth. But for developers, you know, there's stuff like USENIX and so forth there. But for just your average developer or even somebody who's coding as part of a smaller business, they're not necessarily aware of that. So I hate to say it, as an advertising campaign, you know, make the stuff aware, you know, buy some ads on things like hey, you're a developer, here are some things to make your code secure.

Ms. STEVENS. Yes.

Ms. KORAN. Awareness is the biggest challenge of all this, and we've got to kind of market the same way that, you know, all the scam artists tend to do. You know, you can——

Ms. STEVENS. Well, and there's conferences, you know, there's convenings, there's opportunities to do that. And Mr. Behlendorf, what education and training activities is OpenSSF engaging in and how can the Federal Government bolster cybersecurity education in open-source communities?

Mr. BEHLENDORF. Right, well, we've built really 30 hours of courseware available through edX, as well as through the Linux Foundation's own training platform that teach the fundamentals of secure software development. This has to do with everything from a—as your open-source project develops a—you know, a team to respond to security issues to are you fuzzing your code and doing other kinds of testing? And this is content that's taken—been taken about 10,000 people. We'd love to see that more in the hundreds to thousands to millions of people. We're starting to explore partnerships with educational organizations, as well as with companies who might require that of their own developers before they contribute open-source projects on that company's behalf.

Ms. STEVENS. Yes.

Mr. BEHLENDORF. And we're also looking at ways to try to think about if you're choosing between two software packages and one of them, the core developers on that have taken and can demonstrate they've taken this course, you might lean your decision about which software package to use toward the one that has those developers with those credentials associated with it. So we'd love to explore efforts with the Federal Government to——

Ms. STEVENS. Yes.

Mr. BEHLENDORF [continuing]. Expand and, as Amélie referred, to market the availability of these platform—of these—of the courseware.

Ms. STEVENS. Great. Well, thank you. Listen, I'm out of time, but this is obviously a hot hearing, so with that, I'll yield back.

Chairman FOSTER. We will now recognize Representative Feenstra for five minutes.

Mr. FEENSTRA. Thank you so much. I appreciate all the testimony from all our witnesses. It's very impressive.

Mr. Behlendorf, in your testimony you mentioned that very few software developers ever receive a structured education in security fundamentals and often learn the hard way on how their work can be attacked. I'm a bit of an academic myself, and I was wondering what steps need to be taken to address educational gaps for software developers? Is there anything that we can do, you know, to mitigate some of these things up front?

Mr. BEHLENDORF. Well, early in my career I became a fan of a Usenet newsgroup called comp.risks, which was a narrative-focused space where people talked about here's how software has failed and failed in spectacular ways. And it was a very humanizing—some of the root causes for these big failures and understanding why they happened in a way that doesn't assign guilt, it doesn't try to look for, you know, who the bad actors are but simply to say here's where even the best of intentions, the best of processes have sometimes fallen down.

And I think in computer science education, even as we talked about that in elementary schools or high schools, we talk about, you know, the way that developers, most of whom I know are self-taught, come up to speed and start working with different languages, you know, inserting that into all those different kind of informal types of engagements I think is critically important. I think also approaching trade schools, HBCUs (historically Black colleges and universities), other places where, you know, so many of the future work force is being trained is important with—it's not frankly a lot of content. It doesn't require, you know, hundreds of thousands of hours of teaching to get across some of the basic principles of how your software can be used in widely different ways than you might have expected and how to prepare for that, how to take a belt-and-suspenders approach to it, so lots of different answers to that question.

Mr. FEENSTRA. Yes, I appreciate those comments, Mr. Behlendorf. I think you're exactly right. I think there's got to be some parameters, and I also think that people have to be understanding of what they're getting into and the concerns that could occur if they start going down the path and what could actually happen.

Ms. Knausenberger, last summer a JBS support plant in Iowa halted production after it was targeted by a ransom attack. How would improving software supply chain security assist with combating the scourge of ransomware like the attack that hit JBS? What are your thoughts on that?

Ms. KNAUSENBERGER. All right, thank you for the question. So first I'll say, you know, with securing the supply chain, the key steps are know your—where your software came from, do——

Mr. FEENSTRA. That can be a challenge, by the way.

Ms. KNAUSENBERGER. Yes, that can be—it can be a huge challenge. Scanning consistently, doing continuous monitoring in your environment, and above all, keeping your software up-to-date and having processes that allow you to very quickly see what is not up-to-date, what is exploitable. But there are a lot of other attack vectors there that go beyond the software supply chain, and I believe in that case it was a pretty concentrated server attack by a well—a sophisticated adversary. And from what I recall reading in the press, the company did exactly what they should do. They had encrypted backups. They had counsel ready to go and recovered pretty well from that attack.

Mr. FEENSTRA. So I tend to agree. How does the communication get pushed when something like this happens? So are there, you know, packing plants, other things, you know, don't happen. I mean, I'm sure there's a direct source of, hey, we're going to go after these type of organizations. Is there quick communication that can preempt some of this from happening to others once it does happen?

Ms. KNAUSENBERGER. There is a lot of communication between companies that do ransomware negotiations. There are individuals that comb the dark web specifically for—looking for threats. There's a lot of sharing among government entities to help get after this, and of course sharing with companies that might be affected when it happens. But we can't catch everything.

Mr. FEENSTRA. No, absolutely.

I got a quick question, Ms. Koran. In your testimony you discuss how we often look at NIST and the National Cybersecurity Center for Excellence as essential players in interfacing with the open-source community with the guidance and standards they provide. But how—we also need to learn how to lean on them for their new—their methodologies for software assessments and validation. Can you expand just a little bit on this? I've got about 20 seconds left.

Ms. KORAN. Yes. The fact is is they already use this for validating encryption standards and so forth, a similar model of being able for critical software packages to also offer that capability, again, like I said in my statements, is to use that matchmaking capability. It's an awareness issue more than anything, and then to resource that properly so that if we do have critical software packages for the open-source community, make that available to them so they can actually go through that process of assessment and validation.

Mr. FEENSTRA. Wonderful. Thanks for those comments, and I yield back.

Chairman FOSTER. Thank you, and we'll now recognize Representative Tonko for five minutes.

Mr. TONKO. Thank you, Mr. Chair, and good morning to our witnesses. Thank you to the Chairs and Ranking Members for holding this joint Subcommittee hearing. I would also like to welcome and thank the witnesses for being here and sharing information with us.

Open-source software or OSS is not isolated to one nation or to one industry. These projects, as we all know, are ubiquitous, supported by a diverse community of volunteers that collaborate to develop the software through innovation and open discussion. Lucky for us, the United States is home to many of the developers of the open-source software that are used around the world.

As we become increasingly reliant on tech, the Federal Government has an interest in building a closer relationship with the private sector to indeed influence and invest in the security of OSS projects. In our rapidly evolving world, this is an effort that we want to be global leaders in.

Mr. Behlendorf, in your testimony you mentioned that open-source development and security work is inherently international and that several foreign governments are developing their capabilities with respect to OSS. How would you recommend that the U.S. Government take a leadership role in this space?

Mr. BEHLENDORF. Thank you for the question. So the private sector has recently developed an approach among companies for whom software is not their primary function. Organizations like Walmart and Target and Home Depot have all recently opened open-source program offices. These are departments within—sometimes within the IT department, sometimes within legal or marketing, but they are functions within a company that coordinate and help harmonize the engagement of that organization with all the open-source projects that that company might depend upon and even ones that might come out and the contributions upstream that come from that company.

I believe a similar function inside of Federal agencies would help ensure very much a harmonized approach to that. It's very synonymous with, you know, the creation of a CTO (Chief Technical Officer) office within many of the Federal agencies and other things that have spawned out of U.S. Digital Services and GSA (General Services Administration) to be helpful in really building a technical capability inside of different Federal agencies. Finding a way forward for those agencies to know how to engage with the open-source community is really essential.

In my testimony I provided some other suggestions on ways for the Federal Government to engage with open-source projects and approach them. I think the most interesting angle on this is the Federal Government is a major user of open-source software, and, like many other users, is a stakeholder in its success and can approach that as a peer rather than as a—you know, in a top-down regulatory kind of role.

Mr. TONKO. Thank you. I appreciate that. And are there any best practices that our global allies have found to be useful in addressing OSS security concerns?

Mr. BEHLENDORF. In addressing security concerns, the—you know, we've seen countries like France and Taiwan and others recognize the role that the use of open-source software can play in enhancing resilience. And, again, very much it's about funding operations to be able to build that capacity within their own organizations, within those governments to be able to engage with open-source and understand where the risks lie, how to make smart choices about which technologies to adopt, and where potentially to invest.

I do want to note as perhaps an example, as I did in my written testimony, the investment that the State Department made in digital privacy tools over the last 10 years that helped advance the protections for communication in very sensitive areas in a way that helped everybody globally and not just U.S. citizens or U.S. interests. And so, you know, we see—we're seeing that start to emerge from other countries as well, a recognition that there is that collective interest, and actually not even just governments but international organizations like the (WHO World Health Organization) who recently launched their own open-source programs——

Mr. TONKO. Thank you.

Mr. BEHLENDORF [continuing]. In the same way.

Mr. TONKO. Thank you. Do any of our other witnesses care to share any thoughts on this—on these concerns?

Ms. KNAUSENBERGER. I'm happy to share that——

Mr. TONKO. Sure.

Ms. KNAUSENBERGER [continuing]. Through Platform One we are also engaging with allies and sharing code with Five Eyes and across our international community we do do our best to share code functionality as well as cybersecurity concerns and to proactively across governments as well share those cybersecurity concerns.

Mr. TONKO. Sure. And many of you suggested that Federal engagement in open-source will be key to promoting security. So Mr. Behlendorf, can you speak to the cost associated with third-party code reviews and the potential benefits of continuous investment in the top 100 OSS projects? And we only have seconds remaining, so perhaps give it to us in a nutshell.

Mr. BEHLENDORF. You know, the benefit of a third-party code review is having a second or third or fourth set of eyeballs on not just the code you've written but your assumptions and the features and all that kind of thing. And typically an open-source project to do that would cost anywhere from $50-$100,000 to really do this thoroughly, and that seems like a lot of money when you're one open-source developer, but when you realize the impact that we could've mitigated by preventing something like some of the major breaches we've seen, major problems we've seen, it's an infinitesimal amount for the benefit that we would get.

Mr. TONKO. And with that, I thank you and I yield back.

Chairman FOSTER. Thank you. And now we will now recognize Representative Gonzalez for five minutes.

Mr. GONZALEZ. Thank you, Chairman. Thank you to our witnesses and to the other Members. It's very refreshing to have genuine experts amongst the membership of Congress on this issue. It's fun hearing Chairman Foster and Obernolte share their experiences. Dr. Lohn, I want to start with you. As Co-Chair of the AI

Caucus, it's interesting to hear your perspective on how the AI community views open-source software and its perspectives on sharing. I think the benefits are clear, but as you noted, it is imperative to better prepare for potential attacks. In your testimony you noted that if adversaries, particularly China, make the most capable models, then developers in the United States would face an unwelcome choice between capability and security. What are the ways or what is the best way that we can ensure that this does not happen and we're not forced to make that difficult tradeoff?

Dr. LOHN. Thank you for your question. I think that there are a bunch. One of the ways that I would start with is just tracking who—where the progress is across different subfields of artificial intelligence, in language models or in image processing, surveillance, and making sure that we're near the front or at the front in all of those and tracking where progress—where we're falling—where China is gaining on us or where we have a large lead. And then just understanding if we can prioritize which AI systems we are most interested in protecting and understanding which libraries and resources, data sets, and models are the foundations for building those, then we can track which—what the performance benchmarks are for those systems. If we need to, we could provide funding to help support progress in a subfield that's lacking.

Mr. GONZALEZ. Thank you. And then you also noted China is making a push to create programming resources for building systems and the network infrastructure to provide them. If you were to look at it from a competitive standpoint, how close are they to having similar or the same technological capabilities as the United States and/or our allies?

Dr. LOHN. In some fields they're very close. In other fields not so much. In terms of the network infrastructure for distributing, there's a—they're developing their own, but their popularity is way down from where we are. And so in terms of the technical capabilities, that exists. They can distribute the infrastructure. In terms of the popularity, it's still far behind, but that could change depending on how we promote or how we cutoff accesses.

Mr. GONZALEZ. So you said some they're close, others not so much. Which ones are they closer on, and what gives you the most concern when you look at?

Dr. LOHN. The ones that they're closer on are on image processing and in surveillance stuff. They focus a lot on surveillance technology, and so when—in a lot of ways they've pushed the frontiers there. And they've been making a push to be competitive in the biggest, most-impressive areas, which currently are large language models. They've created some models that are very large, which—although they haven't published their performance so we can't do an apples-to-apples comparison.

Mr. GONZALEZ. Thank you. That's all the questions I have, and I yield back.

Chairman FOSTER. Representative Casten will now be recognized for five minutes.

Mr. CASTEN. Thank you so much. I want to echo Mr. Gonzalez. And I am not remotely the computer programmer that our Chairman or Ranking Member are. But I am a big open-source advocate. I—my undergraduate degree was in biology, worked as an entre-

preneur for a while, and I sort of feel like it's a bottom-up versus top-down control question. I'm pro-evolution. I'm pro-free markets for the same reason I'm sort of intrigued by open-source.

However, you know, being opposed to central planning is not the same as being an anarchist. And I wonder if you can sort of stay with that metaphor. My question is for—my initial question is for Dr. Lohn. We are—we've created this AI research—resource research task force as part of the bill we signed—we passed here in 2020. And as you look to their report that's going to come out this summer, as they go through to develop these trusted versions of AI resources that you talked about in your testimony, would you like to see them say there's a protocol by which we identify that these resources are trusted and viable or rather that there is an ecosystem we've evolved that ensures that trusted resources come out of that trusted ecosystem? Because in our markets we have contract law, we have tort law, we have liabilities. We have all the things you need to make a market work. And as you think about how to create an ecosystem that provides trusted AI resources, do you think this is an ecosystem design problem or somebody actually going in and designing the—does that metaphor hold up? Does it—do you have any thoughts on that?

Dr. LOHN. I think it does. I think that metaphor holds up in a lot of different contexts that are relevant to this conversation. Specifically to your question with the AI research resource, I can see both working. I would be even—it would even be a step up for me to just see a label where somebody has come in and said this resource, we know who built it, we have—it has been hashed, we know exactly what it is, and it's got a chain of custody that we understand. That gets an A. This other resource was made by a whole bunch of people that we can't speak to, that gets a C. Even if you don't actually provide the resource, if you just provide a labeling for it, I think that would be a step up.

Mr. CASTEN. It's basically sort of analogous to the sort of scientific research, that we have a peer-review process, and yes, you can do un-peer-reviewed science, but we know how to judge that, right?

Dr. LOHN. Exactly. Something like that would be very helpful. And from the top-down, bottom-up perspective, I think that's a really good analogy, too. Where a lot of the research into what are the vulnerabilities we would be worried about is being done bottom-up. But there—academics often chase the most interesting problem, not the most relevant one, and so I think that government has an opportunity to say these are the problems that we find most relevant——

Mr. CASTEN. And so I want to be quick because I want to get to Ms. Knausenberger before I'm done, but do you think that in this model, you know, the scientific review example I gave, those controls really came from the community? In markets, the controls came from government that set the rules. I don't know how you enforce property rights in the community. Government had to be there. Do you think Congress needs to do more to enforce the rules, or do you think the ecosystem itself is going to evolve these rules on their own?

Dr. LOHN. I'm a little bit worried that the incentives are too far off, that people don't appropriately weigh the impact of a cyber breach until it's too late. And so I would maybe advocate for a little bit of push from the government side.

Mr. CASTEN. OK. So, Ms. Knausenberger, I'm going to ask—lead with a hugely meaty question for you that's unfair in a minute, but be that as it may, the—as I think about these open-source tools in a military context, the rules that we have, if an adversary of ours came and sent a commando unit to disable huge parts of our military equipment, that would be an act of war. We have all sorts of rules; we understand what that means. If they come in and disable the code that runs that military equipment, it's unclear what happens. This is way beyond the purview of the Science Committee, but do we need some kind of a Geneva Convention for cyber warfare? How do we make sure that we have the protections? Because we can put all the rules in place that we want for ourselves, for our country, but I've got to believe you stay up at night wondering about rules that apply in other countries. And we can control our AI, but if China or Russia does something differently, we're beyond the pale. Should we be thinking more about international law in this context?

Ms. KNAUSENBERGER. So, yes, 20 seconds left, you—that is very meaty. I will say that when there is a kinetic effect, it's very clear how we handle things. In the cyber realm, there are some very healthy policy debate right now on where those redlines are, and I think folks well above my pay grade will determine that. But we do take our software supply chain, our cyber posture very, very seriously, and we are investing heavily in this area, especially after—we've had a number of very solid lessons. I guess that's it.

Mr. CASTEN. I guess in——

Ms. KNAUSENBERGER. Unless there's traffic outside the building.

Mr. CASTEN. Well, I guess—and maybe we can follow up off the record, but I'd like—and this may be more for a classified briefing frankly, but I'd like to understand if you had knowledge that a foreign actor came in, interfered with our systems through making changes in our code that destabilized our systems, put—you know, compromised our national security, do you feel that we actually have appropriate recourse that we would have if that was a kinetic event? And if not, how do we protect ourselves? Because it feels like a barn door to me that we've never truly addressed.

Ms. KNAUSENBERGER. I think we should follow up.

Mr. CASTEN. Thank you. I yield back.

Chairman FOSTER. Thank you. And Representative Carey will now be recognized for five minutes. Representative—excuse me, Representative Baird will now be recognized for five minutes.

Mr. BAIRD. Thank you, Mr. Chair. And I really appreciate all the Chairs and Ranking Members putting together this kind of a session. And I always find it very informative when we have experts like the witnesses we have here today to share their experiences with whatever the issue may be.

But—so my question in the area that we really haven't touched on I don't think it is the public-private sector and the partnerships in that area. So how can the Federal Government—and this question goes to all the witnesses. How can the Federal Government,

including the National Institute of Standards and Technology, or NIST, most effectively collaborate with industry and other stakeholders to help secure open-source software? And if we do that, what policy changes can help secure the ecosystem? Ms. Koran, if you want to start.

Ms. KORAN. Sure. I was thinking about the comments I had made in my written testimony regarding the concept of carrier of last resort where, you know, from—the telecom idea is that your local telephone company has to provide you local telephone service, and they are the ones that are there even if—through competition and you have cable companies and whatnot. We have a lot of potentially abandoned software projects out there that are a part of critical infrastructure. They're old. We have to think about like what's out there versus what's to come. And using the government's, you know, scale, size, and matchmaking capability to find potentially abandoned projects for those that are in need of resources and do that matchmaking, you know, NIST can, you know, analyze where, you know, those are going to be most effective within the industries that that's used at but also in more of an agnostic way. You know, they can kind of pick the best experts to kind of assist with that or, again, work with like OpenSSF to, you know, find willing partners to actually, you know, help take over some of those projects as well or at least, you know, find those resources.

Mr. BAIRD. Thank you. Ms. Knausenberger, do you care to comment?

Ms. KNAUSENBERGER. Certainly. I appreciate the question and the previous answer. I will—we are pretty new, I would say, in our open-source journey in the Department of Defense, at least as far as embracing it publicly. We do have a voting membership on the CNCF (Cloud Native Computing Foundation). We are encouraging vendors to come work with us and allowing our software developers to contribute back to the code base as part of their normal coding duties.

I do want to make just an invitation to my fellow witnesses to come and partner with us even more fulsomely. We would love your direct input, and this is something that's very important to me and to our department.

Mr. BAIRD. Mr. Behlendorf?

Mr. BEHLENDORF. I'm always hesitant to comment to any group of people with an open hand and ask for money, and very much I'm not doing that here. The private sector is organizing a series of efforts to try to systematically improve the state of security across the open-source landscape. Mainly the ones I've talked about in written testimony, other groups out there, and all of those groups by themselves, short of the resources they would really like to be able to be as comprehensive as they would like to be. Some of those plans are very specific such as recoding things in the memory-safe languages, as we talked about. Some of them are very broad such as what would it take to fund security audits and remediations of the top 100 or 200 or 500 open-source projects each year? There's a certain degree of scale that government can bring and scale representing the collective interests of all Americans that would really benefit certain key ways. And so I'm happy to explore further those opportunities with all of you and with others in government.

Mr. BAIRD. Thank you. Dr. Lohn?

Dr. LOHN. Thank you. I think there are—so the AI community has been judged as not wanting to work with the government, but that was several years ago, and further studies have found that was drastically overstated, and there is a lot of interest in working with government. I think that there's an opportunity to do more placements, people who come in and work within parts of government, parts of DOD perhaps for short stints of time either part-time or full-time and then go back to their positions. I think that that exchange might be even more valuable than some of the financial opportunities, so creating those positions and opportunities for lateral or temporary transfers.

Mr. BAIRD. Thank you. I thank all the witnesses for sharing with us today. And I've got about 15 seconds left, so I yield back, Mr. Chair.

Mr. TONKO [presiding]. The gentleman yields back his 15 seconds.

Next, the Chair will recognize the Representative from North Carolina. Representative Ross, you're recognized for five minutes, please.

Ms. ROSS. All right. Thank you very much, and thank you for holding this hearing, Chairwoman Stevens, Chairman Foster, Ranking Member Feenstra, and Ranking Member Obernolte. And thanks to all of the witnesses for being here today.

I'm so glad that we're holding this timely hearing, given the prevalence of open-source software and the work being done in my district in this space. I'd like to start my questioning by asking unanimous consent to enter the into the hearing record a statement from Red Hat, a homegrown North Carolina global innovation success story from my district on why we need a holistic approach to software cybersecurity, as embodied in the Administration's cyber Executive order.

Mr. TONKO. And without objection.

Ms. ROSS. Thank you so much, Mr. Chair.

For any of our witnesses, the open-source security vulnerabilities seem, of course, highly concerning. Are there any signs that investments made in response have been yielding results? So, for example, I note that in the Senate Homeland hearing on Log4j, a company in my district, Cisco Systems indicated that they had about five times faster response to Log4j this past year as compared to the similarly widespread OpenSSL Heartbleed vulnerability in 2014. What can we learn from this experience and the vulnerabilities that are both more rare and more quickly remediated in the future? And that's to anybody.

Mr. BEHLENDORF. Perhaps I'll jump in and start. You know, we as humans are very bad at evaluating the longtail risks where something bad that is very unlikely to happen does happen and causes all of us to scramble. We also need to acknowledge that any nontrivial amount of software is likely to have a defect that is yet to be discovered, right? And so there is a constant game afoot at not only enabling innovation, enabling the development of new code, but doing even more to progressively add more controls, more ways of monitoring, more ways of testing to try to find and discover new kinds of vulnerabilities.

No one—nobody here, I think, can sit here and say we've got the key to being able to prevent the next Log4j from ever happening, right? But we do see in things like—I mentioned the Project Zero research that Google had looked at, response times to security vulnerabilities discovered as the beginnings of some econometrics around trying to evaluate that.

We also recently worked with Harvard to publish something called the open-source Census II, which tried to identify what are the most critical open-source projects out there and asked which ones of those have adopted the best practices around security. And we plan to evolve that further into something akin to a dashboard to try to drive a race to the top so to speak amongst the popular open-source projects to adopt more practices and more scanning and more—be more proactive about being able to validate their security research. In fact, most people will tell you if you really do care about security and when you're evaluating open-source software, you look not just for the number of GitHub stars, you know, an indication of popularity, but you look at how many vulnerabilities have been found and fixed because I'd much sooner trust the package that had a lot of holes found and fixed quickly than the one that—in which they haven't yet been discovered or fixed, right?

So we are getting better at this. We still don't have a great, you know, bullet-point metric to be able to illustrate, you know, how much metric progress we're making, but this is an area of active research in our field.

Ms. ROSS. Does anyone else have anything to add? It looks like Ms.——

Ms. KORAN. Yes. Yes, as a former computer security incident responder, it's never a matter of if, it's a matter of when. As you mentioned about the speed of the response is that companies, as they started consuming the software, have plans on how they're going to respond to an event. So that's not just a particular technical looking at the code. That's the integration of your operations teams, the rest of your security teams. That's also part of all of this is knowing what to do when something does happen.

So as part of that investment is looking—again, speaking to the prior Representatives' statements, it's a holistic approach. It's not just one narrow thing, and that's where we actually need to focus on is an entire integration of the response and look at things holistically about how we structure and architect things.

Ms. ROSS. Well, I see my time is about to expire. Thank you so much to the witnesses. Mr. Chair, I yield back.

Mr. TONKO. The gentlewoman yields back. The Chair now recognizes the gentlewoman from Oklahoma. Representative Bice, you're recognized for five minutes, please.

Mrs. BICE. Thank you, Mr. Chairman. First, let me throw this out to all of the panelists today. There has been a lot of discussion about open-source approach versus closed source. Can you expand on the security concerns of open-source and also maybe what are some of the latest techniques that we're utilizing to try to secure it, so, for example, tokenization, or what are we doing to try to secure code currently? I'm happy to——

Ms. KNAUSENBERGER. I'll jump in briefly. So really the same concerns are there whether it's commercial software or open source. But if it's open-source software, you have the power of the crowd looking at it, and then you can also run your own tests internally because it is open code. You can run all of—you can redo the work yourself if you so choose.

With commercial software, you can't see the source code. You do have situations where like with SolarWinds you can have a sophisticated adversary come in, inject malware, and have it be months before anyone knows that there's a problem, whereas in the open-source community we've seen with a number of examples that we just catch it faster, we can push it faster, we have more people trying to fix it faster and spread the word, whereas the commercial side you have some really smart companies working on it but we might not know about it as soon.

Mrs. BICE. Anyone else want to elaborate on that?

Ms. KORAN. Yes, I was going to say as a formal CTO where I led development teams, one of the challenges is the gluing together of stuff. It's not always just one package that's consumed. As I mentioned in my opening statement was that the fact that it is put together, and while they're developed, they can be valid in the state that they're in, but you can't necessarily dictate how they're actually glued together. So as you use multiple packages, you have that particular challenge of them doing a complete systems tests rather than just looking at the code base itself.

And that's one of the challenges here of looking at something as an automated code check that, yes, it may pass those particular evaluations, but it's up to the consumer whether it be an enterprise, an organization, or an individual, to also go through an end-to-end testing. So that's part of that education and part of those capabilities and part of that—those types of services that need to be made available not just to developers but to organizations and industry.

Mrs. BICE. Fantastic.

Mr. BEHLENDORF. Those have been some great answers. I—you know, culturally speaking, there's a greater emphasis on security in the open-source software community. There used to be very much a perspective of, you know, caveat emptor. I'm just throwing this out there, anyone who wants it is welcome to it, and—but buyer beware. And let us know if you find any bugs.

And increasingly we see open-source foundations formalize a structured security team and incident response team within the project itself to in some cases pay for part-time or full-time security researchers who do nothing but try to improve both the underlying code as well as the processes that lead to the development of that software.

Third-party audits are an increasingly important part of release processes. I see many open-source projects that before they release the next .0 version of their software will hire an outside firm to come in and review and audit the code and challenge the things that they found. So it gives me a lot of hope, but there also is a very long tail that is getting longer and longer of very, very small components that, when aggregated together, you know, create interesting things but where there's perhaps less oversight.

You're probably familiar with the phrase that has been around the open-source community a while, with enough eyeballs, all bugs are shallow. Well, we have a problem with a critical number of eyeballs on enough open-source projects, even sometimes highly dependent ones. So one thing we're really trying to do is just make sure that we find the pieces that are critical, find the ones that are under-resourced and where we can direct resources of whatever form are required to increase the level of trust that we might have in that component, that we do so.

Mrs. BICE. Thank you for that. Final question with just a minute left, how is Executive Order 14028 on Improving the Nation Cybersecurity been leveraged to bolster open-source software security? Has there been progress and are there—obviously, there's gaps, but where do those gaps remain?

Mr. BEHLENDORF. I'll jump in on that. It has been tremendously helpful to the folks who've been working on the software bill of materials space for a number of years. Initially, many of the—much of the SBOM activity has been focused on licensing and conformance. In fact, the Linux Foundation has facilitated the development of a standard called SPDX (Software Package Data Exchange), which has become a very widely used standard but used in ways that haven't yet risen to the surface. And what was very helpful about, you know, 14028 was setting the tail end of that process, setting a demand for it that has started to drive that demand upstream to the open-source projects that depend upon it. So we are—we're seeing a shift toward SBOMs. We're seeing it starting to be baked into developer tools and major supply chains as well, so it has been very helpful in driving that ubiquity.

Mrs. BICE. Perfect. My time is expired and, Mr. Chairman, I yield back.

Chairman FOSTER. Thank you. And Representative Perlmutter will be recognized for five minutes of questions.

Mr. PERLMUTTER. Thank you, Mr. Chair. And I came in a little late, so I apologize, but I did hear, Mr. Behlendorf, when you said you were more comfortable if you saw that there had been a number of vulnerabilities found in a particular piece of software, you felt more comfortable about that than one that didn't have so many found. And that just was counterintuitive to me. Can you elaborate on that little bit? And, Ms. Knausenberger, maybe you can because you talked about SolarWinds and the commercial versus open-source.

Mr. BEHLENDORF. Yes, I should be—I should clarify found and fixed, not just found and left in their current state because that indicates people care. That indicates people are looking and scrutinizing it. And I've very rarely seen software that, you know, you can use for any nontrivial purpose or amount of time that hasn't had a defect found in it. So it's a sense of health of a project that you find—that people are looking for them, finding them, and that the project's own developers acknowledge those bugs and are ready to fix them. That is a—I much prefer that over the thing that looks and feels perfect and done and no bugs.

Mr. PERLMUTTER. OK.

Ms. KNAUSENBERGER. Yes, that was a great answer, and I will agree. If there are no bugs found in a particular piece of software

it's because no one's looking. It's not because it's perfect. And it is a sign of pedigree to have lots of eyes using and really poking away at a piece of software, so it also engenders confidence to me.

Mr. PERLMUTTER. And you may—all of you may have answered this before, but who is it? Who are those eyes? Who—what—who is the community? I mean, are they doing this for grins? Do they get paid? How—who is it that's looking to see if there's a vulnerability?

Ms. KNAUSENBERGER. So the top four contributors are Microsoft, Google, Red Hat, and Intel. And the Department of Defense is increasingly involved as well looking and contributing back. There are people that are paid to also contribute to open-source software, and I'm sure my fellow witnesses can jump in on that as well.

Mr. PERLMUTTER. I mean, more often than not it's not my nephew who's a super computer nerd, you know, sitting in the basement looking to hack something?

Ms. KORAN. Sometimes you trip over it. As part of, you know, building software systems, you may find that the code that you are intending to use between versions or, you know, a new use of it creates an unexpected result, and that could be a bug or a vulnerability that is found. And hopefully, based on licensing or, you know, the policy of an organization, those changes do get floated back into the original code base. Sometimes it doesn't just because of—you know, sometimes the sensitivity of intellectual property of the organization that has actually integrated that, so it does run into conflict sometimes with the licensing. But others—you know, obviously, we do have a very robust security community that does, for grins and giggles, you know, goes to like poke at software. And the same thing that we use to secure it from the development standpoint, fuzzing and other tests, are also used by adversaries, but they're just not reported.

Mr. PERLMUTTER. OK.

Mr. BEHLENDORF. I'd like to——

Mr. PERLMUTTER. Go ahead. Somebody else wanted to——

Mr. BEHLENDORF. I'd just like to add that, you know, studies have found since the earliest days of open source that the vast majority of contributions that come into any major open-source project come from developers who are using that code to solve a business problem, right? In some cases, it's to train themselves up to further their careers, to get a bit of notoriety, or it might even be that nephew in the basement who has a brilliant idea that becomes the basis for the next cool thing. And the great thing is those—all those interests and all those agendas can align and harmonize and create interesting and innovative code. The key is finding processes, processes that allow for more than one set of eyeballs to look at code before it's released, processes that test and vet for off-by-one errors that lead to memory vulnerabilities and the like. So from motivation's point of view, it really is industry and individuals working voluntarily together but to solve real-world problems.

Mr. PERLMUTTER. I may not be able to get this answer in before my time expires, but what role do you think open-source coding software has in modernizing our electrical grid as we add more and more renewables if anybody can answer that?

Mr. BEHLENDORF. I'll jump in and say, you know, at the Linux Foundation we have a project called LF Energy, which is a collaboration between some of the major grid operators around the world and hardware and software providers to them to develop the next set of software infrastructure for—totally focused on renewables and microgrids and enabling a much greener kind of future. And so we're seeing active involvement from industry on that, but we've also seen the solar community working on open-source software to tie into grids to manage resources in a really explosively cool kind of way and really excited to be leading that project.

Mr. PERLMUTTER. Thank you. My time is expired. I yield back to the Chair.

Chairman FOSTER. Thank you. And Representative LaTurner will now be recognized for five minutes of questions.

Mr. LATURNER. Thank you, Mr. Chair.

Ms. Knausenberger, you discussed in your testimony how the Iron Bank repository can reduce supply chain risk. Can you elaborate on Iron Bank's levels of security and provide insight into whether non-DOD entities are able to replicate this effort or security components of it?

Ms. KNAUSENBERGER. Certainly. So, first, the way that we secure our open-source software leveraging Iron Bank, we start with the onboarding process where we validate the supplier identity of the code. We then harden. We scan for vulnerabilities and dependencies. We do policy configuration and scanning. We ensure that we are doing automated updates and have a process to pull those in. We do delivery and auditing as well, and there are a lot of things that come into that process as well as an SBOM. So those are the things that we are doing within Iron Bank, and we do that—some of that work is done organically. Some of that work is done through partners and—largely commercial partners.

As far as private citizens or the public leveraging Iron Bank, it is available in itself and an open-source product, and we have again heard of at least one commercial bank leveraging Iron Bank and a variety of defense contractors and other interested parties.

Mr. LATURNER. Thank you. Ms. Koran, in your opinion, would cracking down on waste, fraud, and abuse in Federal agencies like HHS, for example, help mitigate cybersecurity threats by disincentivizing bad actors?

Ms. KORAN. It could. Unfortunately, it's a resourcing issue again. I remember being on some of the security mailing lists and whatnot, and usually at a—most of these health organizations, whether it be hospitals or, you know things—places where grants are made to you, it's one IT person doing multiple roles. So, you know, sometimes it's not necessarily out of malice, but it's also out of available resources.

But also one of those cases is—too, is, you know, even at HHS, which is the biggest IG (Inspector General), the cybersecurity oversight was a really, really small group out of the entire organization, so it's also a scaling issue just to be able to provide that level of oversight. It could help crack down on it, but it needs definitely a lot more resourcing. And I'm sure that's the case with many of the other agency OIGs.

Mr. LATURNER. Sure. You stated in your testimony that triaging software vulnerabilities is hampered by a lack of technology literacy among the people in charge of responding to such events. In your opinion, how should the government fix these miscommunications to speed up future response times?

Ms. KORAN. Well, for one is definitely staff experts like the rest of the panel here in places where they can actually do the most good. My time at OMB, most of those folks were not technical even in the CIO's office, so, you know, educating them correctly and I had to explain, you know, the overflow and what that actually meant in terms that were meant, you know, for folks who are non-technical. That level of literacy needs to be higher up in the food chain. And while that can happen as, you know, our generations start to take over more, you know, as some of the older generations retire and we move up, but, you know, in this case we can't wait. So in some cases it's just a matter of pairing the right people in the right place at the right time.

Mr. LATURNER. Sure. I'd appreciate that.

Dr. Lohn, in your testimony you discussed how AI system models coming from our adversaries may back American software and AI developers into a corner. Can you explain what you mean when you say the United States would have to choose between capability and security?

Dr. LOHN. Yes. If the—the best model—so, all right. Someone when someone is building a new AI system, they rarely design it from scratch. That can cost millions of dollars. And so what they'll often do is download a model at a starting point and then use a small amount of data to tweak it. But that—when they're trying to decide which model to start from, they choose the most powerful one that exists or near to it. And if that—currently, most of those are American-made or some of our allies, but if that were made by an adversary, then you would have to choose between taking this one that wins on all the benchmarks or taking the 10th-place model or the 2nd-place model and using that as your starting point. If the—that most powerful model is an adversary-made one, they can embed triggers in it that they can use later on after the model has been retrained for its new purpose.

Mr. LATURNER. Thank you very much. Mr. Chairman, I yield back.

Chairman FOSTER. Thank you. And I'd also like to mention for those Members who are attending either in person or virtually that we're going to attempt a second brief round of questions if we have time.

And we'll now recognize Representative Meijer for five minutes.

Mr. MEIJER. Thank you, Mr. Chairman.

I want to touch a little bit on the conversation that Mrs. Bice had a little bit earlier with Dr. Lohn. And it seems like one of the themes that's coming out here is this tension between, you know, government shifting from an approach of propagating best practices to then expanding and setting some mandatory minimums within kind of Federal supply chains and Federal—or Federal kind of IT infrastructure and that the tension between the voluntary approach and more of a mandate or restrictions or certain minimums. You know, from the NIST voluntary framework from the 2014 Ex-

ecutive Order 13636 for critical infrastructure, you know, through to that White House E.O. 1428 from last year, last May for minimum standards for Federal systems.

Are there—you know, we're kind of looking at the government's role vis-à-vis protecting and encouraging updates of—whether it's the software bill of materials to get that full kind of codification of what's in the stack on the open-source side. But so much of the perpetual vulnerabilities—and this came across in several of the testimonies—are just the failure to update even after an exploit is known, even after a patch is issued and how that can take a very long time. You know, the Log4j patch deployment took—you know, identified a week—you know, kind of take weeks to months, potentially years to be fully onboarded.

Are there—apart from the government, apart from some of the kind of industry and then nonprofit organizations that are represented here—and I just want to throw this out to the group—you know, what role do—specifically with the commercial sector, what role do insurance providers who may be on the hook for a breach that results in data being compromised, has a financial impact, or, you know, investment firms? I mean, are there other entities that should be part of this conversation, apart from just government, nonprofit, and some of the larger kind of commercial IT partners?

Mr. BEHLENDORF. I'll jump in. I certainly believe that the insurance industry has a—could have an important role to play, and we have not yet them really seen—really—yet really seen them show up to the discussion. You know, insurance is how we brought greater degrees of safety to, you know, our transport systems, to so many other parts of modern society. Many organizations do offer cybersecurity breach insurance and other kinds of risk insurance related to this, and they've had difficulty in finding premiums and pricing, as I understand it, to actually make for sustainable models.

This is tied somewhat to the challenges we have in applying metrics to understanding just how secure is this bundle of software that we deployed or this cloud service that we're using as an enterprise. But with better metrics, with better monitoring we can get, I think, to a model that works for the insurance industry. So I would be really eager not only to engage with them but engage with many of you in government who oversee many of those activities to try to figure out how to encourage more bridge-building there.

Mr. MEIJER. Thank you, Doctor. Anybody else want to offer input?

Ms. KORAN. Yes. Yes, definitely one of the challenges here, especially going back to Representative Bice's comment there, was that with a lot of the standards and everything, everybody looks at these as the high watermark, as the thing that they're going to only meet and meet there. But resilient systems, whether they be in critical infrastructure or your everyday business down the street that's consuming this needs to build above that waterline, above that mark so that if and when an event does occur, that they are resilient enough to sustain that and maintain their business and operations. And that's the biggest impact to global economies is that resiliency thing.

You know, looking at the Colonial Pipeline, looking at things such as SolarWinds, because, you know, the base was never met, you know, when they were hit, everybody scrambled and everybody panicked. And that's one of those cases of using that to drive better behavior in industry is to look to go above and beyond, just use that as that baseline to move rather than the roof.

Mr. MEIJER. Well, and certainly I also serve on the Homeland Security Committee, and I think, you know, while we're focusing on the role of NIST, you know, CISA's role is also very critical here. But I think the—Ms. Koran, the point is very well taken on the high watermark and building above that line because we can only look back to what has occurred and have to be focusing as well on what's coming down the pike.

So with that, Mr. Chairman, I yield back.

Chairman FOSTER. Thank you. And at this point I think we have enough Member interest for an additional round of questions. So I'll begin by recognizing myself.

And my general question is what's the future of automation in code verification? You know, if 10 years from now or 20 years from now or six months from now are we going to actually be able to have most of these packages verified by some piece of code that you run on them when you attempt to update them? And what's the sort of state-of-the-art, and what your guess for the anticipated state-of-the-art over the next few years? Yes.

Mr. BEHLENDORF. I'll jump in on that. Right now, our current what are called static analysis tools and even the dynamic analysis tools, the ones that look at running systems and try to look for vulnerabilities, these tools suffer quite a bit from what are called false positives. You know, developers who use them have to wade through a lot of signals that aren't really about vulnerabilities for the one or two pieces that actually do indicate some problems. So there is a ton of research now going into how to reduce those numbers of false positives, how do you make this tool perceived as productive and useful to developers as they work for that code, and then how do you turn that on at scale?

We've done quite a bit at the OpenSSF in partnership with many of our members around looking at fuzzing, which is a way of throwing a ton of garbage data at a piece of software to understand how and when it might break, looking at ways to try to automate that at scale across thousands or millions of projects at once. It is a very hard problem, but it is an area of active research in this space.

I also think that the—as—the better we get at finding new ways to automatically detect and remediate vulnerabilities, the more kinds of new vulnerabilities we'll discover because the landscape always shifts. And what we need is a general capacity for not just finding and fixing bugs but finding new kinds and fixing new kinds of bugs. And that's why I—you know, what we call for in terms of investments really is on a persistent basis rather than, OK, now, we have security in our source codes and we can all go home.

Ms. KORAN. And I was going to throw something in there, too, is you give somebody a problem and there's going to be, you know, 10 million different ways to solve it. For large programs and open-source packages, you know, you can enforce standards of people, how the code—different things that they need to do, but the won-

derful thing about open source is that anybody can basically go into it regardless of their skill level or their experience and time doing this.

So with AI, part of that—those models are going to be necessary to be checked, we can do some level of automation, but you go to someplace like Stack Overflow, throw out a question, and you're going to have 100 different answers how to implement the exact same thing. Whether or not they're correct, some of them may be, some of them may not be. So as Brian mentioned about fuzzing, we will only catch a certain amount through some of that automated testing, but it will still require some eyes and that rigor of being able to understand that you should code in a certain fashion. Providing core repositories of known checked methods and implementation standards are ways to do that, but until everybody's kind of on board or willing to follow those models, that's the challenge of humanity is, you know, you have that level of creativity, want to kind of do it their own, so I think that's going to be our biggest challenge moving forward.

Dr. LOHN. I would also just like to very quickly pipe in and say that there's an opportunity—there's a lot of focus on vulnerability discovery and patching. I think that the patching side, as the previous Congressmember mentioned, is a big problem where there's a bigger opportunity to make a push. It's a harder problem, but there's more opportunity for gains.

Ms. KNAUSENBERGER. All right. So the macrolevel, we're already pretty automated across the community, even in the Department of Defense. Pretty much anything that we are building in the last few years has an automated pipeline. We also are really big on bringing in the hacker community after the fact to hack away at our systems and tell us maybe what we missed in production as well and in our development environments. But even there when we see a particular play used more than once, we want to look at, well, how do we automate that play so that, as we test this on the backend, we can become increasingly creative. We don't have to just do the things that we tested before. We do the same thing in our pipeline as we noticed that maybe we missed something. How do we give that feedback by adding a new tool, giving feedback to a vendor, et cetera.

But automation plays heavily. I'd say we're more nascent in the way that we can automatically check algorithms, and that's just as a—you know, as an industry. But I think that there will be a lot more automation leveraged there in the future as we make just more advances.

Chairman FOSTER. Yes, did any of these tools flag Log4j?

Ms. KNAUSENBERGER. So I think the key thing to understand there is that if you're using a scanning tool, it's not necessarily going to identify malicious code as much as it's looking for code that's going to break something or is incorrect or is a bug. And so a lot of vulnerabilities—it's not because the code was, you know, inherently wrong. It's that someone was very clever and they found a way to exploit something that was working in the code to do ill. And that's why it's so important that we continue to have the cyber research community engaged and the people that are using that code for business purposes engaged because some of these things

you're not going to find until you start hammering away at it no matter how good you are.

Chairman FOSTER. My time is up, and I'll recognize Representative Obernolte for five minutes.

Mr. OBERNOLTE. So let me start with Dr. Lohn. I'd like to continue a line of questioning that Congressman LaTurner had started. You were talking about the tradeoffs between capability and security and discussing the possibility that a competitor such as China might introduce a very capable piece of open-source software in the hopes that it would be incorporated along with perhaps an embedded security vulnerability into sensitive software. So my question for you is how would one avoid that? Because as you've also said in your testimony, it can be very difficult, especially when you're talking about software-related AI. It's going to be very difficult to determine whether or not a vulnerability exists even if it was put there intentionally. So what could we—what can we do to solve that problem?

Dr. LOHN. That's a very good question. The first thing we could do is make sure that we have superior or competitive models of our own that we do trust. Short of that, trying to discover these—what they—are sometimes called backdoored or Trojanized models is challenging. NIST has an effort right now called TrojAI where they're trying to run competitions with all of—with many backdoor models where academics or researchers are trying to find new ways. It's—that would be—being able to detect whether a model has been Trojanized would be great. It's the ambitious solution, and I think that we should not put our eggs all in that basket. Trying to create our own competitive models is my primary suggestion and then creating other like diplomatic or bureaucratic means for gaining trust would be my second.

Mr. OBERNOLTE. So just tunneling down on that, though, wouldn't—we're competing—we're introducing competitive models that don't have those potential security vulnerabilities. I mean, we—that's difficult to do. You can't just monitor the marketplace for open-source software and every time a capable module is introduced from a question—from a competitor, develop a better module. That's not what you're suggesting, is it?

Dr. LOHN. What I'm suggesting is that we prioritize the AI applications that we're interested in and then make sure that we have the talent and incentives to stay at or near the cutting edge in those particular areas.

Mr. OBERNOLTE. OK. Yes, I think this is something we should talk more about because I think you've highlighted a very interesting and important potential problem.

And, Ms. Koran, if I could ask a question of you, we have discussed through a couple of different lines of questioning today the Administration's Executive order from last year. And the consensus has been that it was a positive development. But in your written testimony you had a somewhat different reaction to it. You called it a dark cloud over the agencies and you feared it might, to quote you, "stifle innovation and self-determination and put a chill over industry." So I wanted to give you an opportunity to give your point of view on that.

Ms. KORAN. Yes, definitely. So one of the challenges, you know, having been a Federal CTO, was a lot of these times is it's usually you're already kind of doing the work that you're doing and then you get an Executive order or demand from another agency like CISA to go and do a thing, and that actually requires you to change—you know add a reporting capability to it, do some extra checks. You may have been doing some of that already, but the idea is is now you have to comply.

Then agencies, either through acquisitions through GSA via the Federal Acquisition Service and so forth when you acquire software and services, it puts an onus on industry whether or not they want to comply, and it reduces the ability for agencies to kind of pick and choose from a better menu.

One of the challenges about FedRAMP, which is the cloud security compliance side of things, that's a very limited marketplace because it's such a high bar for a lot of companies to reach, and most of the ones who are innovating are small companies. Most of the ones you usually see in FedRAMP are ones who've had the time and the money to get there. So it does stifle innovation because it does remove the lack of choice and availability of software and services to agencies to work their mission.

Mr. OBERNOLTE. Yes, interesting. Well, I think you've highlighted a very important topic, which is the bond of trust that has to exist between government and the open-source community because, you know, without that bond of trust, neither of those communities can do their jobs. So—and, you know, as you've expressed, the government's efforts in other areas to provide this kind of assistance have not been crowned with glory, so I appreciate the viewpoint. It's something we'll have to work on.

Mr. Chair, I yield back.

Chairman FOSTER. Thank you. And we will now recognize Representative Perlmutter for five minutes.

Mr. PERLMUTTER. Thanks, Mr. Chair. And the Chair sort of prompted a question, as did the Ranking Member talking about trust. So what happens if either in the original software or somebody from the online community turns out to be a bad actor, you find malicious software that appears to have either been—was intentionally placed there to be triggered at some later date? What happens, one, if you find that? What happens if you find somebody in the open-source community is trying to create trouble with some sort of malicious effort? What happens to those bad actors?

Mr. BEHLENDORF. Well, I'll share, you know, there's an incident in 2020 where a research team at the University of Minnesota decided to test that very question and see whether the Linux kernel community would notice when they submitted a bug—like a software patch that had a backdoor embedded inside it. The patch came in, it started to work its way through the process, and within about five days the developers—the maintainers on the Linux kernel noticed the bug, noticed that it was intentional, responded by blackballing basically the entirety of the University of Minnesota IP (Internet Protocol) address space from ever contributing again to the open-source—to the Linux kernel.

That might have sounded extreme, but the community of well-run open-source projects have not only processes to detect these

kinds of things that no matter how many tools you use, you still have to boil down to humans looking at what's coming through and evaluating and trying to understand what's really going on inside of the software source code but also have strong social mores against that kind of contribution.

Now, there are other open-source projects without those kinds of processes, without even a lot of developers involved, sometimes modules that are really just written by one or two people. And in that—and in recent cases such as—might have heard of colors.js, fakers-js, IPC.gov. There have been modules written by one person or a small number of people, and, as I understand it, they've all been either Americans or Europeans in these recent examples where they've decided they would use that privileged position they have to put something in. And some of those got noticed very quickly. The ones that inserted a cryptocurrency miner do tend to get caught pretty quickly because they drive your CPU (central processing unit) crazy. But it's—that's much—it's—that's I think a space we have much less of a systematic solution for except to say we should prefer those components that are built by teams rather than those components built by individuals.

And I think you'll see—start to see more of that work its way into supply chain validation processes as well over time. I'll use this component that has more eyeballs on it, more positive attestations to the integrity of that software than this other piece that comes from an individual no matter what IP address range, what company, what country. You know, let's look at the substance of what's been created.

Mr. PERLMUTTER. You mentioned crypto in your—in that answer. So I don't know how many thousand cryptos are out there now, types of currencies, but is there some—is the Treasury, is—you know, we have Defense here, but is somebody looking to see if those cryptos have some sort of malware in them, either intentional or not? I'm just curious whether that can create problems if we start taking cryptocurrencies generally as some sort of payment.

Mr. BEHLENDORF. The answer is it's—yes, a lot of people look at that because it automatically has built into it a bug bounty where in many cases if you're able to find a vulnerability in a major cryptocurrency platform, you get rewarded with the ability to mint new dollars or transfer funds to yourself. And so one thing we've found is while there have been a lot of famous kind of hacks and things, compromises, that's a community that also perhaps takes security more seriously than many other parts of the open-source community because the stakes are so much higher.

And I think if we're looking for a space where zero trust is really a first principle, it's very much in that community. But it should absolutely be in our mind as we think about central bank digital currencies, the use of distributed ledgers for supply train traceability, and these other kinds of opportunities that technology does provide.

Mr. PERLMUTTER. Thank you. My time is about to expire. I'm going to yield back to the Chair. And thank you for this hearing, Mr. Chair and Mr. Ranking Member.

Chairman FOSTER. Thank you. And before we bring this hearing to a close, I want to thank our witnesses for testifying before the

Committee today. The record will remain open for two weeks for additional statements from the Members and for any additional questions the Committee may ask the witnesses.

The witnesses are now excused, and the hearing is now adjourned.

[Whereupon, at 12:06 p.m., the Subcommittees were adjourned.]

Appendix

Additional Material for the Record

LETTER SUBMITTED BY REPRESENTATIVE BILL FOSTER

GitHub

88 Colin P Kelly Jr Street,
San Francisco, CA 94107
Tel: 415-448-6673 (main)

The Honorable Eddie Bernice Johnson
U.S. House of Representatives
Committee on Science, Space & Technology
2321 Rayburn HOB
Washington, DC 20515

May 10, 2022

Dear Chairwoman Johnson,

Thank you for the opportunity to provide GitHub's perspective for the joint hearing of the House Committee on Science, Space, and Technology, Subcommittee on Investigations & Oversight and Subcommittee on Research & Technology, on "*Securing the Digital Commons: Open-Source Software Cybersecurity.*"

GitHub is the largest code repository and home to the largest developer community in the world with 83+ million developers, 4+ million organizations, and 200+ million repositories on our platform. GitHub is where most of the world's software development happens, and as such, we are uniquely positioned to help secure the open source software ecosystem.

Today, 99% of codebases contain or depend on open source. Many of the services and technology we all rely on, from banking to healthcare, depend on open source software – and open source, like all software, can have vulnerabilities. Open source software is often developed on a voluntary basis and individual project maintainers do not always have the same resources or dedicated security teams as the businesses that rely on their code. That's why we see open source security as a responsibility and challenge we need to address together as a community, and there are roles for industry, government, organizations, and the developer community to play in creating solutions.

Our role at GitHub is to help developers and maintainers realize improved security outcomes. We do this through building tools and processes that support developers to code securely throughout the entire software development lifecycle and by continuing to invest in building security functionality into the platform.

Empowering developers with training, funding, and tooling

Securing the open source ecosystem starts with empowering developers and open source maintainers with tools and best practices that are instrumental to securing the software supply chain. GitHub offers open source developers access to training, funding, and tooling that help secure open source software, including:

Training
GitHub Security Lab offers free security training and educational materials to developers on subjects including tooling (CodeQL, Fuzzing), defensive programing, and security best practices. Security Lab researchers also audit open source projects for free and partner with project maintainers to help them fix security vulnerabilities.

Funding
GitHub Sponsors enables the community and businesses to financially support the people who design, build, and maintain the open source projects they depend on, with millions of dollars flowing to open source projects across the globe every year.

Tooling
To scale the security of open source, developers must be able to prevent security vulnerabilities before they happen. That is why GitHub builds security features and tools into developer workflows so that security is addressed earlier in the development process. For developers, and particularly those with limited security experience, these tools reduce friction and lower the barrier to entry, enhancing the security of the overall ecosystem. Our tools and features help address:

- ***Securing dependencies***. The most common security vulnerabilities in most apps are the result of using vulnerable open source dependencies. GitHub provides a comprehensive suite of features that monitor the National Vulnerability Database for vulnerabilities, alert developers to vulnerable dependencies, and automate remediation, resulting in higher remediation rates in less time.

- ***Securing new code***. GitHub provides a static analysis solution called CodeQL which can detect vulnerabilities in first-party code as it's created. CodeQL does this using an open source set of queries that find vulnerable coding patterns.

- ***Credential leaks***. One of the most frequent security mistakes we see is the accidental inclusion of credentials into source code. GitHub secret scanning protects developers by scanning, notifying, and automatically revoking leaked credentials before they can be used maliciously.

Preventing and addressing malicious attacks

The previous section addresses issues related to vulnerabilities in code – 99.8% of which are introduced by accident by well-intentioned maintainers. While malicious attacks make up just 0.2% of the vulnerabilities GitHub found in our customers' dependencies, they grab outsized attention.

Malicious attacks on open source security usually occur when an attacker deliberately introduces code that they intend to exploit, generally through some form of account takeover. Hardening account and code security is critical to prevent these attacks. Strong authentication, including the use of two-factor authentication (2FA), as well as commit signing and package immutability are essential to protecting the software supply chain and securing the software development ecosystems. Industry funding and engagement has a meaningful impact, particularly for package registries that run on a voluntary basis. GitHub is committed to protecting developers and their projects with the

usable, strong authentication options and we are requiring 2FA for all users who contribute code on the platform by the end of 2023.

The role of government in securing open source software

Like the private sector and the developer community, the government also has an important role to play in securing the open source ecosystem. To that end, we share the following thoughts on areas where the government can contribute:

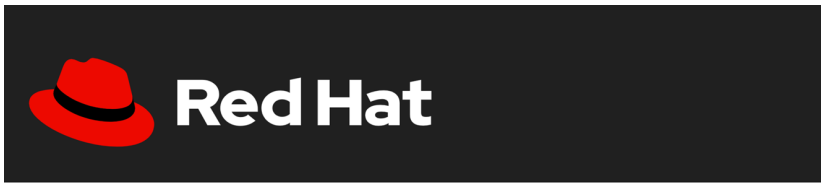
- Identify the open source software that is critical to the Federal government and the Nation's critical infrastructure, and prioritize funding to safeguard the security and resilience of open source software and its supply chains. This investment could be made in partnership with open source foundations, such as Open Security Software Foundation (OpenSSF), that specifically focuses on open source security.
- Direct and fund the National Institute of Science and Technology (NIST) to engage the open source community (including open source foundations, platforms supporting open source, open source projects, and users of open source) in the development and review of secure development and supply chain risk management guidance and standards. Such engagement would help ensure that guidance and standards can be adopted equally by proprietary and open source software. In addition, research projects could use guidelines on how to create sustainable open source communities that outlive the research grant.
- Request that NIST review the National Vulnerability Database (NVD) and make recommendations to improve its support for automation (e.g. incorporating structured software identifiers such as package-url), improve actionability (e.g. incorporate exploitability data such as VEX), and investigate integration with Software Bills of Materials (SBOM).
- Make cybersecurity education on secure software development fundamentals broadly available. This should be incorporated in broader efforts to improve cybersecurity education and training for all IT professionals.

At GitHub, we believe that the security of open source is critical to the security of all software. As the home to the world's largest open source developer community, GitHub is uniquely positioned and committed to helping developers advance the security of their code and we take this responsibility seriously. We look forward to working with the Committee and Congress to advance this important work.

Sincerely,

Stormy Peters
Vice President, Communities

LETTER SUBMITTED BY REPRESENTATIVE DEBORAH ROSS



May 10, 2022

The Honorable Bill Foster, Chairman
The Honorable Jay Obernolte,
Ranking Member
Subcommittee on Investigations and
Oversight
U.S. House of Representatives
Washington, DC 20515

The Honorable Haley Stevens, Chairwoman
The Honorable Randy Feenstra,
Ranking Member
Subcommittee on Research and
Technology
U.S. House of Representatives
Washington, DC 20515

Dear Chairman Foster, Chairwoman Stevens and Ranking Members Obernolte and Feenstra:

Red Hat, Inc. ("Red Hat") appreciates the opportunity to provide our views on efforts to improve the security, trustworthiness and resilience of all the software we depend on, prior to the Subcommittees' hearing on May 11, "Securing The Digital Commons: Open-Source Software Cybersecurity." We ask that this letter be included in the hearing record.

We commend the Subcommittees' recognition that there are risks associated with all software -- open-source software as well as proprietary -- and that there are unique advantages that open source can provide, e.g., in terms of innovation, enabling a new generation of digital transformation, and enhancing cybersecurity.

Regardless of the software development model, policy makers' focus must remain on promoting effective software development and life cycle management practices as embodied in the May 2021 Executive Order on Cybersecurity. In that context, open source provides numerous advantages to provide trust and resilience, if the user has the right procedures in place to know the software they are using, monitor risks and vulnerabilities, and make updates and fixes, as appropriate.

Below, please find further elaboration on our views.

601 Pennsylvania Avenue, NW Suite 900 North Building Washington, DC 20004

ABOUT RED HAT

Red Hat is the world's leading provider of open source software solutions using a community-powered approach to deliver resilient and high-performing cloud, Linux, middleware, storage and virtualization technologies. As a key to our success, we deliver these solutions with transparency and accountability from start to finish.

Unlike many software publishers (and users) who often see 'community' upstream code as an easy, cost-minimizing 'grab and plug in' to their products, Red Hat curates upstream community source code for review and inclusion into Red Hat products which undergo extensive composition, security screening, and quality assurance testing prior to release. All packages are digitally signed and distributed through resilient channels.

Through a continuous, dedicated process to support our customers Red Hat tracks developments on product components, assesses implications of identified vulnerabilities for their impact on users and customers, and continuously provides security updates, using the approach outlined in an open approach to vulnerability management.[1]

Red Hat engineers actively engage in a wide range of community projects, including serving as maintainers in some instances. Red Hat actively contributes its code improvements to the open source community upstream for the benefit of all developers and users.

THE CYBER EO FOCUS ON SOFTWARE LIFECYCLE MANAGEMENT IS CRITICAL

The Executive Order on Improving the Nation's Cybersecurity ("Cyber EO")[2] established software security as a national priority. It set out a comprehensive approach to ensuring that software used in critical missions, regardless of its development model, is trustworthy and resilient.

The core tenets of the Cyber EO remain fundamental to improving the security posture of all software — **both proprietary and open source** — including assuring that vendors of all stripes maintain greater visibility into their software, take responsibility for its life cycle, and make security data publicly available. Absent the focus on open and transparent software management over the life cycle of a software product, as envisioned by the Cyber EO, little progress will be made.

---

[1] With regard to Apache Log4j, as an example of our on-going support, Red Hat provided our customers with regularly updated information (including what products were NOT affected) and tools to enable them to identify whether the issue was present.

[2] Executive Order 14028, "Improving the Nation's Cybersecurity", May 12, 2021, found at: https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/.

The Cyber EO was a much needed, appropriate response to serious incidents involving software vulnerabilities such as WannaCry, Hafnium, and SolarWinds — none of which involved open source software.

In response to the Log4j incident, the Senate Committee on Homeland Security and Government Affairs,[3] held a hearing in February to review the incident and other issues related to software security. At that hearing, witnesses repeatedly emphasized that what Log4j, Solarwinds and other incidents illustrate is that we are faced with a software security challenge that does not discriminate between proprietary and open source. At that hearing, Brad Arkin, SVP and Chief Security and Trust Officer at Cisco, went a step further noting that it would be a mistake to single out open source as a unique source of risk. Mr. Arkin observed:

> *"It is … incorrect to assume that open source software is uniquely a source of risk. All software has the potential to contain vulnerabilities. All software used in enterprise and commercial products and services require lifecycle management."*
>
> *"Together we need to further improve baselines for software security, including open source software. We collectively need to improve our speed and efficiency at finding and fixing problems when they arise. And together we need to boost our resilience against attacks, particularly as we work to develop, distribute and apply software patches and mitigations."*

In his testimony, Dr. Trey Herr, Director of the Cyber Statecraft Initiative at the Scowcroft Center of the Atlantic Council, similarly established that what we are faced with is a software security issue that is not unique to open source:

> *"Log4j is not exceptional. Software supply chains both open source and proprietary have been victim to and remain vulnerable to widely exploited flaws."*
>
> *"The track record of software is one of insecurity; no different for open source and proprietary code. And so our discussion should today look beyond the intricacies of a single incident."*

Likewise, Jen Miller-Osborn, Deputy Director of Unit 42 at Palo Alto Networks, brought the point home during the Q&A:

> *"You know, this isn't an open source problem. This is inherently that software will have vulnerabilities regardless of whether or not it's proprietary or open source."*

---

[3] "Responding to and Learning from the Log4Shell Vulnerability", hearing before the Senate Committee on Committee on Homeland Security and Governmental Affairs, February 8, 2022, found at: https://www.hsgac.senate.gov/hearings/responding-to-and-learning-from-the-log4shell-vulnerability.

For these and many other reasons, Red Hat continues to applaud the Administration for its comprehensive approach to software supply chain security, as embodied in the May 2021 Cyber EO. We commend, in particular, the crucial role of the National Institute for Standards and Technology ("NIST") for its essential leadership in providing the practical roadmap of this critical endeavor. A continued, dedicated focus on implementation of the Cyber EO and NIST guidance, and the objective of openness and transparency, is essential.

ENHANCING THE SECURITY POSTURE OF UPSTREAM COMMUNITY OPEN SOURCE

Not all open source software (nor for that matter all proprietary code) is the same. It is best to understand the open source development model as an ecosystem, which varies by code quality, scale and purpose. Key to understanding this ecosystem is that software components are often consumed directly from an upstream community source or from a 'warehouse'[4], potentially pulled as a dependency, and not from 'downstream' community distributions or enterprise products.

Many vendors see upstream source code as a quick, cost efficient way to incorporate components into their product offerings. The software end user, especially in the enterprise and federal IT environment, is also key to improving the overall exposure risk and enhancing our nation's cybersecurity posture. The Cyber EO's focus on improving the overall posture of software (whether proprietary or open source) directly addresses that changing this approach is key.

- Thus, effective implementation of baseline security standards for development of software sold to the government (**Section 4 guidance**), including requiring developers to maintain greater visibility into their software and making security data publicly available, is essential.
- Increasingly, the challenges facing all software publishers now include more than traditional vendors. Digital transformation in the enterprise (and in government) means that all companies and organizations are becoming software companies. The discipline required to produce robust,

[4] Many developers get their 'open source bits' from software distribution 'warehouses' such as, for example, the Python Package Index for python software. Engagement with these key centers of distribution, which fill a vital function, to improve their security posture could reap significant ecosystem benefits for software developers, vendors and users. This is consistent with the comprehensive approach on supply chain. Many of the packages and much of the community code found there is unsigned and has little information on its provenance .

The Atlantic Council, concluding that when properly managed, open source software may be more secure, has identified along with others who have studied the software supply chains that these 'warehouses' are a prime vector for those looking to introduce and exploit vulnerabilities, more so than efforts to try to compromise individual 'upstream communities'. See "Breaking trust: Shades of crisis across an insecure software supply chain", July 26, 2000, especially discussion around Fig. 3, found at: https://www.atlanticcouncil.org/in-depth-research-reports/report/breaking-trust-shades-of-crisis-across-an-insecure-software-supply-chain/.

resilient software-based solutions requires everyone to improve their software use and production processes (independent of OSS or proprietary vendor development).

- Federal IT leadership should consider greater education on the benefits and possible incentives for agencies to utilize commercially maintained open source solutions as a default for essential missions. Increasingly, given the complexity of threats, relying entirely on self-support may not be sufficient in today's more complex world.
- Recognize that the days of 'grab and plug in' software components can no longer be tolerated. Attention must remain on meaningful software development and lifecycle management, including dedicated response, recovery, discovery and resilience programs. This is a key thrust of the Cyber EO, which includes significant steps to achieve these objectives in the federal IT context.
- Moreover, the software end user must make it a priority to be aware of, download, and update software components when those bugs are discovered and a fix is available.

In the context of this holistic approach, there are several initiatives Red Hat wants to highlight that focus on 'upstream' open source software, for example:

- Collaborative initiatives such as **OpenSSF**, hosted by the Linux Foundation ("LF"), create significant opportunities for broad scale improvements in the quality and resiliency in open source community development. Red Hat is a founder in that effort. We encourage a wide swath and diversity of the software industry to participate.
- Future work should examine and identify the tools community developers and maintainers need (and currently do not have) to better identify and track vulnerabilities and enable software development and lifecycle management practices.
- One notable initiative under the sponsorship of the LF, is the **sigstore project**. Led by software industry leaders including Red Hat, Google, HPE, Cisco and VMWare in collaboration with Purdue University, sigstore seeks to improve the security of the software supply chain by enabling the easy adoption of cryptographic software signing backed by transparency log technologies. This will enhance the provenance utility of community source code. The service will be free to use for all developers and software providers, with the sigstore code and operation tooling developed by the sigstore community.

To restate the key policy point: these initiatives, as well as others, are important. But without effective adoption of recognized software development and lifecycle management processes by vendors, users and organizations, they alone cannot solve our challenge.

FOSTERING AN OPEN AND TRANSPARENT SOFTWARE ECOSYSTEM

A key element to improving the overall posture of software security is transparency and openness. The Cyber EO outlines concrete steps which the federal IT procurement process needs to effectively implement and enforce.

With open source an informed user can be made aware of an issue prior to fixes being applied, giving opportunity to mitigate. With proprietary software a user often has to wait for and apply a patch. A proprietary software vendor might not acknowledge, issue a notice for, nor fix a Low or Moderate/Medium vulnerability whereas with open source software (because it's all public) — especially where there are products supported by a trusted vendor — that knowledge will be available and customers can make appropriate risk-based decisions in advance, or absence, of a patch.

As serious consideration is given to the topic of the hearing, we urge the Committee to take into account the following fundamental principles:

- Strong cybersecurity involves focus on the trust and resilience of all software, regardless of its development model.
- As embodied in the Cyber EO, a risk-management approach that promotes comprehensive recognized software development and lifecycle management practices is fundamental.
- The open source ecosystem is dynamic and diverse. The success of any initiative depends on collaborative engagement, openness and transparency.
- Any recommendations, especially at the technical or practice level, must be unencumbered and freely usable.

Thank you, in advance, for your consideration of our views. Please let us know if you have any questions or can provide additional information.

Best,

Mark Bohannon
Vice President, Global Public Policy
& Associate General Counsel