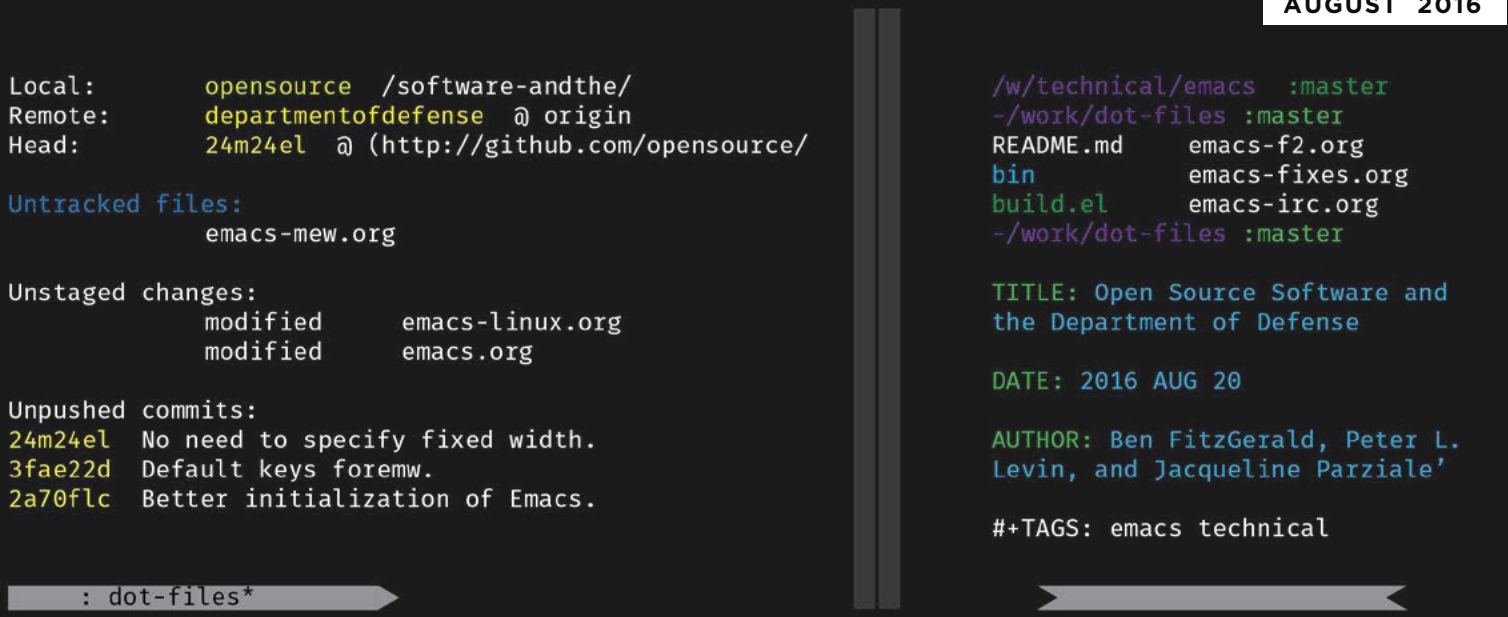
AUGUST 2016



OPEN SOURCE SOFTWARE & THE DEPARTMENT OF DEFENSE

Ben FitzGerald, Peter L. Levin, and Jacqueline Parziale

Foreword by
General Hugh Shelton, U.S. Army (Ret.)

Center for a New American Security

About the Authors

Ben FitzGerald is the Director of the Technology and National Security Program at the Center for a New American Security. His work focuses on the intersection of strategy, technology, and business as they relate to national security. Recent projects have included analysis of the future of the global defense industry, U.S. military technology superiority strategies, cyber security collaboration, and innovation within the Department of Defense. Prior to CNAS, FitzGerald founded the U.S. subsidiary of an Australian strategy firm, Noetic, leading their work with the Pentagon, military services, and the U.N. Earlier in his career, he worked with information technology companies IBM, Rational Software, and Unisys. His work and commentary have been featured in the media from C-SPAN to *Vice*.

Dr. Peter L. Levin is an Adjunct Senior Fellow for the Technology and National Security Program at the Center for a New American Security. He is the founder and CEO of Amida Technology Solutions, an information technology firm that focuses on data and data security. He is also a consulting professor in the Department of Aeronautical and Astronautical Engineering at Stanford University, and a fellow at the Beeck Center for Social Impact & Innovation at Georgetown University. From May 2009 until March 2013, he served as Senior Advisor to the Secretary and Chief Technology Officer of the Department of Veterans Affairs, where he led their health-record modernization, open government, and open source initiatives.

Jacqueline Parziale is a researcher with the Technology and National Security Program at the Center for a New American Security (CNAS), and was previously the Joseph S. Nye, Jr. Research Intern for the same program. Prior to joining CNAS, she was the Research Editor and Project Coordinator at Rowan Technology Solutions, where she developed *The West Point History of Warfare* and *The West Point Guide to the Civil Rights Movement,* and the Fellowship Coordinator for the Madison Policy Forum's Military-Business Cybersecurity Fellowship program.

Acknowledgements

The authors would like to thank the Defense Advanced Research Projects Agency (DARPA), particularly Wade Shen, and Giant Oak, particularly Gary Shiffman, for making this paper possible. They would also like to acknowledge the help of Dan Geer, the Chief Information Security Officer at In-Q-Tel. The authors are especially grateful to General Hugh Shelton (Ret.) for his kind foreword.

Readers should note that CNAS receives funding from some of the organizations mentioned in this report. CNAS maintains a broad and diverse group of more than 100 funders, including private foundations, government agencies, corporations, and private individuals, and retains sole editorial control over its ideas, projects, and products. Red Hat, Inc. does not provide financial support to CNAS. A complete list of financial supporters can be found on CNAS' website at cnas.org/content/cnas-supporters.

The views expressed in this report are those of the authors alone, who are solely responsible for any error of fact, analysis, or omission.

About the Technology and National Security Program

The Technology and National Security program explores the nexus of strategy, technology, and business to develop practical ideas that capitalize on opportunities and mitigate risks associated with the rapid pace of technological change. The program advances new ideas associated with cybersecurity, identifies innovative ways to generate technological advantage, and builds better connectivity between the commercial sector and the U.S. national security establishment. Leveraging strong relationships across government, technology, and industry organizations, the program brings together disparate stakeholders to help ensure emerging technologies are effectively employed to support U.S. national security strategy.

Foreword

General Hugh Shelton, U.S. Army (Ret.)

I first heard of open source in 1994. I was on a ship off the coast of Haiti commanding the U.S. Joint Task Force en route to reestablish democracy there. Our mission was to remove General Raoul Cédras and his puppet government and return the democratically elected President Jean-Bertrand Aristide back into the country.

Everything was going well for us militarily until the White House asked me to send them information in a specific format. Unfortunately, the proprietary software we were using did not support the format they desired. My team contacted the vendor for help and, predictably, they refused to modify the software to help us.

This is a story that is all too common in the public sector. Simple software changes are impossible, or expensive in time and dollars. Some vendors think they own you forever.

Fortunately, my team didn't stop there. An airman used open source software to get the information in the format the White House wanted, and we were able to deliver it on time. That's when I realized I needed to learn more about open source. It enabled us to do what we need to do to meet requirements.

Fast forward to today, the U.S. Department of Defense (DoD) is one of the largest consumers of open source in the world. Every tactical vehicle in the U.S. Army runs at least one piece of open source software.

Although I'm pleased to see the widespread consumption of open source software in the DoD, we can do so much more. We need to go from simply consumers of open source software to using open source principles as a way to do business in the DoD. Open source development models are generally better than their proprietary counterparts. This is because they can take advantage of the brainpower of larger teams, which leads to faster innovation, higher quality, and superior security for a fraction of the cost.

Open source development methodologies also enable teams to be more nimble and respond faster to the threats our nation faces. Our adversaries are often made up of small, decentralized cells that adapt quickly to new situations. Doing business the old way may hold our warfighters down or even lead to loss of life. This needs to change.

This paper discusses where the DoD is already making inroads to use open source development methodologies to shorten time to mission without sacrificing national security. I hope our civilian and uniformed leaders will read it carefully and take it seriously. There is so much more we can do. We all agree that our warfighters deserve the best equipment our nation can offer. Open source software is an under-appreciated way of achieving that goal.

General Hugh Shelton is the current Chairman of the Board at Red Hat, Inc. and served as the 14th Chairman of the Joint Chiefs of Staff.

IMPORTANT DEFINITIONS

In national security circles, open source software is often confused with other "open" initiatives, which sometimes leads to misunderstanding and incorrect conclusions.

Open Source Software is defined by the Department of Defense 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 that software."[1] Like nearly all software development methodologies, open source software is built collaboratively. For open source software, sometimes that collaboration is private, and sometimes it is public.

Open source software is distinct from:

Open Source Intelligence – intelligence collected from publicly available sources such as the media, social networking sites, government reports, or academia, as opposed to intelligence collected from covert or clandestine sources.

Open Architectures – hardware and software system architectures designed to make extensions, maintenance, upgrades, and component exchange easier. Open architectures enable software developers to create new features and users to easily install them.

Open Data – data that is freely available to use, reuse, and redistribute without restrictions from copyright, patents, or, sometimes, the need for privacy control mechanisms.

Introduction

Senior leaders across the defense establishment are justifiably concerned about the erosion of U.S. military technical superiority and have recently launched several high-level initiatives to ensure continued military advantage for the United States. Secretary of Defense Ashton Carter has been outspoken about the need to collaborate with, and recruit talent from, Silicon Valley and other like-minded technology hubs to increase innovation within the Department of Defense (DoD). Deputy Secretary of Defense Bob Work has been tasked by Secretary Carter to lead the development of a new approach, the third offset strategy, to extend U.S. military technical advantages.[2] The acquisition reform efforts set forward by House Armed Services Committee (HASC) Chairman Mac Thornberry highlight similar concerns and goals.[3]

The capabilities that underpin the United States' current military advantage – such as global command and control; intelligence, surveillance, and reconnaissance (ISR); precision munitions; global positioning systems (GPS); and cyber capabilities – are largely based on information technologies and ultimately depend on the quality of the software upon which they are constructed. From game-changing weapons to routine back-office systems, the DoD is entirely reliant on its ability to identify, acquire, certify, deploy, and manage software.

This challenge maps directly to the fierce competition in the commercial market for technology. Successful organizations in every sector must use all the tools at their disposal to gain advantage, especially as the pace of breakthroughs accelerates and their availability increases. In recent years, the private sector has become increasingly reliant on open source software, which underpins critical software infrastructure from enterprise applications to smartphones and advances from artificial intelligence to electric cars. But while the commercial world has installed repeatable and scalable frameworks that improve the software it uses, the DoD struggles to keep pace. Unless the Department is able to accelerate how it procures, builds, and delivers software, it will be left behind.

From massive delays in the production of new weapons systems caused by software integration issues, to major cost overruns for ordinary projects such as logistics support systems, to the seemingly unknowable cost of its total software portfolio, the DoD struggles mightily to acquire, create, and maintain its software, both open source and proprietary.[4] Indeed, the challenges of software management led to a specific requirement in the 2013 National Defense Authorization Act that the DoD inventory its software use.[5]

Unfortunately, software development is not currently a high-profile, high-priority topic in the discussion

about diminishing U.S. military technical superiority. It should be. And one of the most effective ways the U.S. could maintain dominance and further strengthen system capability is to insist on openly architected systems – as articulated in Chairman Thornberry's acquisition reform efforts – and to make much better use of open source software.

Software development is not currently a high-profile, high-priority topic in the discussion about diminishing U.S. military technical superiority. It should be.

Of course, the challenges of effective software management are not unique to the DoD; the entire U.S. government struggles with such investments. One way the White House is seeking to address these problems is through the increased use of open source software across all federal agencies. On August 8, 2016, the White House Chief Information Officer (CIO) released a Federal Source Code Policy that calls for new software to be built, shared, and adapted using open source methods to capitalize on code that is "secure, reliable, and effective in furthering our national objectives."[6] The policy requires that "new custom-developed source code developed specifically by or for the Federal Government to be made available for sharing and re-use across all Federal agencies ... [and] Federal agencies to release at least a portion of new custom-developed Federal source code to the public"[7]

The draft policy contains a notable exemption for "source code developed for National Security Systems."[8] This exemption is specific. It includes systems that involve intelligence activities, cryptologic activities, command and control of military forces, and weapons, and explicitly excludes systems used for routine administrative and business applications.[9] It would be easy for the DoD to broadly exploit these exemptions in favor of custom proprietary code, but such action would be counterproductive and unwise. The Department should instead seize this opportunity to make greater use of open source methods and more fully embrace the use of open source software. In doing so, it will gain the common mode benefits of open source platforms and methods, as well as important advantages specific to the DoD's needs.

There are many instances in which the Department successfully uses open source software, from the platforms that power Predator drones to DARPA's Memex, a search tool for the dark web. However, at present, the Department is failing to institutionally exploit many best practices available to ensure the optimal generation and management of its code base. While the value of open source software is important to acquisition reform and cost-efficiency goals, its most consequential contribution is to the very foundation of the nation's military capability. The DoD must overcome bureaucratic hurdles and embrace open source software as a critical element of its efforts to maintain military technical superiority in the 21st century.

Open Source Software in the World

Over the past 50 years, open source software has evolved from a process of collaborative convenience for a small community of researchers and academics into a collection of widely recognized and globally adopted software libraries. Open source software powers the Internet, mobile technologies, multibillion-dollar corporations, and even the International Space Station.

Open source software is ubiquitous. Open source software released through the Apache Foundation accounts for almost half of active web servers around the world.[10] The Android operating system, which runs on over 80 percent of all smartphones in the world, is based on the open source operating system Linux.[11] As of 2012, open source software served 75 percent of the top 10,000 websites on the Internet.[12] Open source has been embraced by for-profit businesses – 78 percent of companies use open source software substantially, and only 3 percent don't use open source software in any way.[13] Major corporations such as Verizon and General Electric also take advantage of it. IBM contributes to existing open source projects, releases formerly proprietary code under open source licenses, and creates open source platforms from which it sells other IBM products and services.[14] As Peter Levine of the venture firm Andreessen-Horowitz wrote in 2014, "without open source, Facebook, Google, Amazon and nearly every other modern technology company would not exist."[15]

However, those outside of the software development community often do not understand open source software and its impact, despite its overwhelming proliferation. Opponents cite the potential for information security risks or breeches if mal-intentioned actors have access to source code. Some critics are apprehensive about the sustained profitability of business models built around open source software.[16] Others are concerned

about the challenges of community building and management. The very nature of open source methods makes them less controllable – and therefore less predictable.

Time and again, open source frameworks have overcome these challenges. Information security concerns have been debunked, because increased public scrutiny of code has led to identification and reconciliation of problems that were not discovered through "closed" quality checks. Further, "closed source [versions of] products like Microsoft have been riddled with security flaws and issues," some of which were significant zero-day exploits of widely used, commercially

'Without open source, Facebook, Google, Amazon and nearly every other modern technology company would not exist.'

available products.[17] New business models harness open source methods as a means of increasing both supply and demand for new technologies. GitHub, an online code management repository and collaboration platform and the largest host of open source code in the world, has been valued at $2 billion.[18] Google and Facebook contribute to open source projects for products such as data centers on which they openly compete, and in 2014 Tesla Motors released all of its patents "in the spirit of the open source movement, for the advancement of electric vehicle technology."[19] These decisions hinge on the desire to develop their fields and the recognition that secrecy hurts invention, profits, and security more than it helps them.While not all open source projects prove self-sustaining, those with the right problem set and management structure achieve critical mass, just like proprietary systems.[20]

Open source methods are not the optimal software solution in all instances. However, when used appropriately, they add value in several vital areas:

- First and foremost, open source software can provide **better technical outcomes**. When more people participate, the result is higher-quality code, which is essential for extensibility and scalable platforms that can grow over time. Larger numbers of contributors are more likely to identify flaws within source code and also provide a greater ability to implement solutions to those flaws more quickly.
- Open source software can also provide **significant business advantages**. It has a lower total cost of ownership and lessens the cost of software development not only due to decreased licensing fees, but also because it leverages the work of the community, often at little to no expense.[21] It is cheaper and easier to reuse existing software than to build new software from scratch, and open source software avoids dependency on a single vendor employing closed proprietary products.[22] In addition, open source software allows organizations to capitalize on software systems that are larger and more robust than anything they could create themselves.
- Open source methods also enable **more effective collaboration**. Indeed, generating value from collaboration is the fundamental premise of open source software. Not only do open source methods optimize the technical quality of software, but they also allow disparate groups of developers to work together. This is particularly valuable for software projects that have large, diverse, and distributed sets of users. Effective collaboration allows open source methods to move entire fields of research forward while creating benefits for individual organizations, communities, and society at large.

The LinuxWorld conference and expo convened members of the open source software community from around the world. (Wikipedia)

The open source software community has proven itself as an essential element of the software development landscape and the technological infrastructure that underpins a digital world.

MISCONCEPTIONS ABOUT THE SECURITY OF OPEN SOURCE SOFTWARE

Discussion of open source software in national security is often dismissed out of hand because of technical security concerns. These are unfounded.

To debunk a few myths:
- Using open source licensing does not mean that changes to the source code must be shared publicly.
- The ability to see source code is not the same as the ability to modify deployed software in production.
- Using open source components is not equivalent to creating an entire system that is itself open sourced.

As In-Q-Tel's Chief Information Security Officer Dan Geer explains, security is "the absence of unmitigatable surprise."[23] It is particularly difficult to mitigate surprise with closed proprietary software, because the source code, and therefore the ability to identify and address its vulnerabilities, is hidden. "Security through obscurity" is not an effective defense against today's cybersecurity threats.

In this context, open source software can generate better security outcomes than proprietary alternatives. Conventional anti-malware scanning and intrusion detection are inadequate for many reasons, including their "focus on known vulnerabilities" that miss unknown threats, such as zero-day exploits. As an example, a DARPA-funded team built a flight controller for small quadcopter drones based on an open source autopilot readily downloaded from the Internet. A red team "found no security flaws in six weeks with full access [to the] source code," making their UAV the most secure on the planet.[24]

DoD's History with Open Source Software

The DoD uses open source software successfully, but infrequently and on an ad hoc basis. There are many examples of individual agencies or teams using open source software in their projects including, in the Fiscal Year 2017 IT President's Budget Request, projects ranging from a tool that automates aviation mission planning tasks to a database designed to manage the holdings of the U.S. Army Historical Collection.[25] In fact, groups within the DoD open source their projects so often that the National Security Agency created its own GitHub page and DARPA developed its Open Source Catalog.[26] The current open source codebase used throughout mission critical systems across the public sector and in the DoD demonstrates its legality and technical, operational, and economic success.

The proliferation of open source software inside the Pentagon is a strong counterargument to many, mostly passive objections against open source software in military-specific situations, and ultimately is irreversible. Indeed, a 2003 MITRE report found 115 applications of open source software in the DoD, concluding that "banning FOSS [free open source software] would have immediate, broad, and strongly negative impacts on the ability of many sensitive and security-focused DoD groups to defend against cyberattacks."[27]

The MITRE report could validly make this bold case based on the current breadth and depth of DoD use of globally available open source software from programmer toolkits such as OpenMap to the Apache web server. Take, for example, the proliferated software component called OpenSSL. This widely used open source library implements the "Transaction Layer Security" (TLS) protocol, which is the basis of secure communication on the Internet. "TLS connections are everywhere on the Internet," explains Dan Boneh, a leading researcher in applied cryptography and professor of computer science and electrical engineering at Stanford University, "and by far the most widely used TLS software is open source."[28] In other words, the most critical part of the network infrastructure – the security layer underlying all communication – is secured by open source software.

The proliferation of open source software inside the Pentagon is a strong counterargument to many, mostly passive objections against open source software in military-specific situations, and ultimately is irreversible.

The underlying issues associated with the use and adoption of open source software are best seen in DoD policy statements from the last several years. The DoD has made occasional attempts to address ongoing

misunderstandings of what open source software is and how it can be gainfully deployed.[29] In 2009, then-acting DoD CIO David Wennergren published a memorandum, "Clarifying Guidance Regarding Open Source Software," in which he enumerated the advantages of open source software and explained the existing policies surrounding its execution in an effort to clarify "misconceptions and misinterpretations of the existing laws, policies, and regulations" that had "hampered effective DoD use and development of OSS," or open source software. [30]

Senior Pentagon leaders are currently focused on developing innovation, maintaining technological superiority, and managing costs, as outlined in policies such as the DoD's "Better Buying Power" series. The DoD has an IT budget of over $38 billion for FY17, and given the

U.S. Airmen from the 163rd Maintenance Group, which is a part of the California Air National Guard and primarily involved in the Predator missions, train on a ground control station. The Predator drone runs on the open source operating system Linux. (Val Gempis/U.S. Air Force)

amount of code it develops for programs from weapons systems to travel-booking systems, the Department can naturally benefit from proven open source advantages such as reuse and code quality.[31] But despite this, there has been very little to no discussion of how the DoD can systematically incorporate open source software into the products it develops, and no specific mention of open source methods or licensing in programs of record in recent years.

Defense-Specific Use Cases of Open Source Software

Considering the DoD's top-down apathy toward and difficulty with using open source methods, one glaring question remains: Why is there continued bottom-up support for open source software and methods within the DoD? In addition to universal benefits, open source software contributes distinct value to the Department, given its mission and organizational constraints. The following are six clear cases in which open source software provides specific advantages to the DoD.

Better Platform To Build To

The DoD and its industry partners build a significant amount of specialized software, from simple mobile apps for training to the Distributed Common Ground System, the U.S. Army's intelligence fusion application.[32] Leveraging open source platforms such as operating systems, databases, middleware, and toolkits allows the DoD to deploy and maintain software more quickly and flexibly than proprietary alternatives. For example, the Persistent Close Air Support system developed by DARPA runs on Android. General Atomics drones, including the Predator and Reaper, and ground stations operate on Linux, a switch that was made after Windows-based systems proved vulnerable to malware such as keyloggers, software that captures a user's keystrokes.[33] Deploying custom applications on open source platforms has the potential to allow the DoD to more rapidly develop, utilize, and deploy military-unique software at lower cost.

Simpler Interagency Collaboration

The opportunity to improve software across all federal agencies is at the heart of the White House's Open Source Software Policy initiative. Because they have fewer proprietary restrictions, or in many cases none at all, open source systems are easier to use when multiple agencies need to collaborate or share code seamlessly. The nature of the DoD's mission means it is exposed to the most demanding technical challenges and threats. Concomitantly, the DoD is better resourced to develop technology than virtually anywhere else in government. Sharing quality software with the rest of the U.S. government would support the DoD's cyber mission to "defend the homeland and U.S. national interests against cyberattacks of significant consequence."[34] For example, the NSA recently open sourced the Systems Integrity Management Platform, a security compliance toolkit, to help others avoid duplication and to improve the quality

of their own products.[35] The interagency could share other government-sensitive tools internally using open source methods, which would make it easier for the DoD to share code improvements, simplify current collaboration obstacles, and ensure its systems stay relevant. This is true both within DoD agencies and across other departments with which it collaborates, such as the Departments of Treasury, Justice, and Homeland Security, as well as other law enforcement agencies.

Better Codevelopment of International Systems

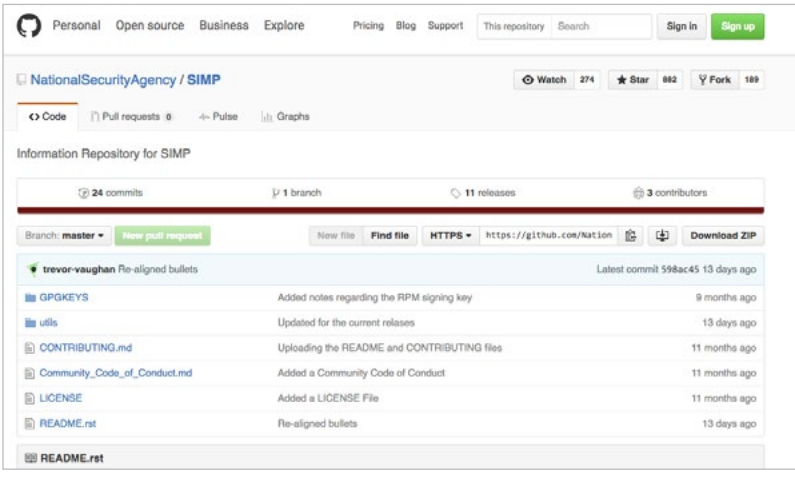
Foreign military sales (FMS) and codeveloped military platforms are increasingly important to the Department of Defense for alliance-building and economic reasons. Many DoD platforms create physical architectures that implement alliances and make them real; this is particularly true for command, control, communications, computer, intelligence, surveillance, and reconnaissance (C4ISR) systems and weapons systems intended for use in allied operations. Additionally, with growing costs for defense systems and declining budgets among many U.S. allies, nearly all nations must collaborate to jointly develop new weapons systems. This can be seen most acutely with the Joint Strike Fighter, which has nine partner countries, all of which develop components for the aircraft.[36] Effective collaboration on source code improves quality, creates more binding ties between alliance members, and allows for more rapid upgrades to enhance operational efficacy and provide security updates.

FMS and codevelopment projects are made more complex by export controls. The elegant use of open source licenses could obviate the need for International Traffic in Arms Regulations (ITAR) licenses in some instances, or otherwise create more flexible use and improvements to software once licensed.

Simpler Reuse and Accreditation

Accrediting software within the DoD is both important and laborious.[37] To the extent that accredited software components or tools can be reused, the Department could benefit from the time saved not only in software development but also in the accreditation process, which can often be lengthier than development itself. It is this realization that is at the heart of the methods behind efforts like DARPA's xDATA project, which provides techniques and software tools for processing and analyzing large and incomplete data.

A soldier uses a tablet with the Android operating system, which is based on open source software, to direct an airstrike. (DARPA)



The NSA shares its Systems Integrity Management Platform (SIMP) software on GitHub, a repository for open source code. (NSA)

Increased Competition

Because the underlying mechanisms of open source systems are shared, new vendors can build on or alter existing systems more easily than under proprietary constraints. This ability enables more robust competition for programs that must interface with existing military systems as well as for upgrade and maintenance projects, a key goal of the Better Buying Power series. Similar goals are behind the open systems architectures language that is featured prominently in the acquisition reform efforts set forward by HASC Chairman Thornberry.[38] Open source methods would further help achieve the HASC's goals.

Expanded Innovation Opportunities

In the same way that organizations such as Google and Facebook benefit from open sourcing code in order to extend their capabilities, the DoD can gain advantages from external collaboration on challenging projects. DARPA's xDATA project seeks to develop "computational techniques and software tools for processing and analyzing large, imperfect and incomplete data."[39] As Gary M. Shiffman, CEO of Giant Oak and professor at Georgetown University, said while cautioning against enthusiasm for aggregated data in the absence of software to perform the needed analytics, "We frequently hear the analogy that data is the new oil. But what good is unrefined oil? Analytics is the new refinery."[40] Solving this problem in the defense context will also have positive non-defense implications, thus encouraging the broader community to contribute to and prosper from collaborative software development. In addition to benefiting generally from a more advanced field, the DoD can also create classified, military-unique branches of its code for particular purposes, and can do so without sharing those changes back to the open source community.

In the same way that organizations such as Google and Facebook benefit from open sourcing code in order to extend their capabilities, the DoD can gain advantages from external collaboration on challenging projects.

Open source also provides the opportunity for government research and development to support the growth of commercial applications. The NSA released Niagarafiles (NiFi), a dataflow automation tool, and Accumulo, a secure data store, via the Apache Foundation in order "to move technology from the lab to the marketplace, making state-of-the-art technology more widely available and aiming to accelerate U.S. economic growth."[41]

MISCONCEPTIONS ABOUT OPEN SOURCE SOFTWARE AND MILITARY ADVANTAGE

One of the most important objections to open source defense systems is the apparently deliberate squandering of technological advantage. Some fear that if DoD source code is readily available to U.S. adversaries, those actors may be able to use it to their advantage. This apprehension is misplaced.

Irving Wladawsky-Berger, a former IBM senior executive credited for organizing its industry-leading efforts around the Internet and open source, explains that as hardware platforms and software systems become increasingly commoditized, "the real differentiation is in the data, and in the expertise you need to manage and extract insights from the data." [42]

Similarly, the United States does not derive its military technical superiority from source code, but from the effective integration and adaptation of its doctrine, organization, training, materiel, leadership and education, personnel, and facilities (DOTMLPF). Software is a vital enabler of U.S. military capability, but it is the configuration of and the data housed in these systems that provide advantage, not the source code itself.

Impediments to DoD Use of Open Source Software

In spite of these clear benefits, the DoD continues to fail to fully capitalize on the opportunities presented by open source software. The primary hurdle is cultural – the DoD is a large bureaucracy, open source methods, though widely used in industry and even in the defense establishment, are not considered standard practice inside the Pentagon, and change is hard. Erroneous and unfounded misunderstandings about open source software create confusion and concern, and lead busy project managers, acquisition professionals, and senior executives to simply avoid open source as a viable option for DoD software projects without serious consideration. This generic cultural malaise can also be attributed to four specific, and addressable, impediments.

Lingering, Unfounded Concerns about Access to and Visibility of Code

In spite of clear evidence to the contrary, many defense professionals continue to believe that the use of open source software licenses means that adversaries will see and manipulate the code used in DoD systems.

Differing Management Philosophies

There are very real human and cultural factors that create a perceived mismatch between open source methods and DoD acquisition. The Pentagon is a hierarchical organization, which naturally leads to top-down methods of technology management through formal requirements. This approach tends to be risk-averse and stifles experimentation, innovation, and rapid implementation. In contrast, open source grows without direct supervision or control, and relies instead on collaborative, bottom-up development of solutions to problems.

Erroneous and unfounded misunderstandings about open source software create confusion and concern, and lead busy project managers, acquisition professionals, and senior executives to simply avoid open source as a viable option for DoD software projects without serious consideration.

Acquisition Hurdles

While there is no specific prohibition against the use of open source software, the requirements, accreditation, and contracting steps required to get an open source software project through the acquisitions process can be a challenge to navigate. Further, the DoD's contracting methods are not designed to facilitate the services-related purchase and customization of open source software. The acquisition system also provides a convenient scapegoat for those disinclined to use open source software for other reasons.

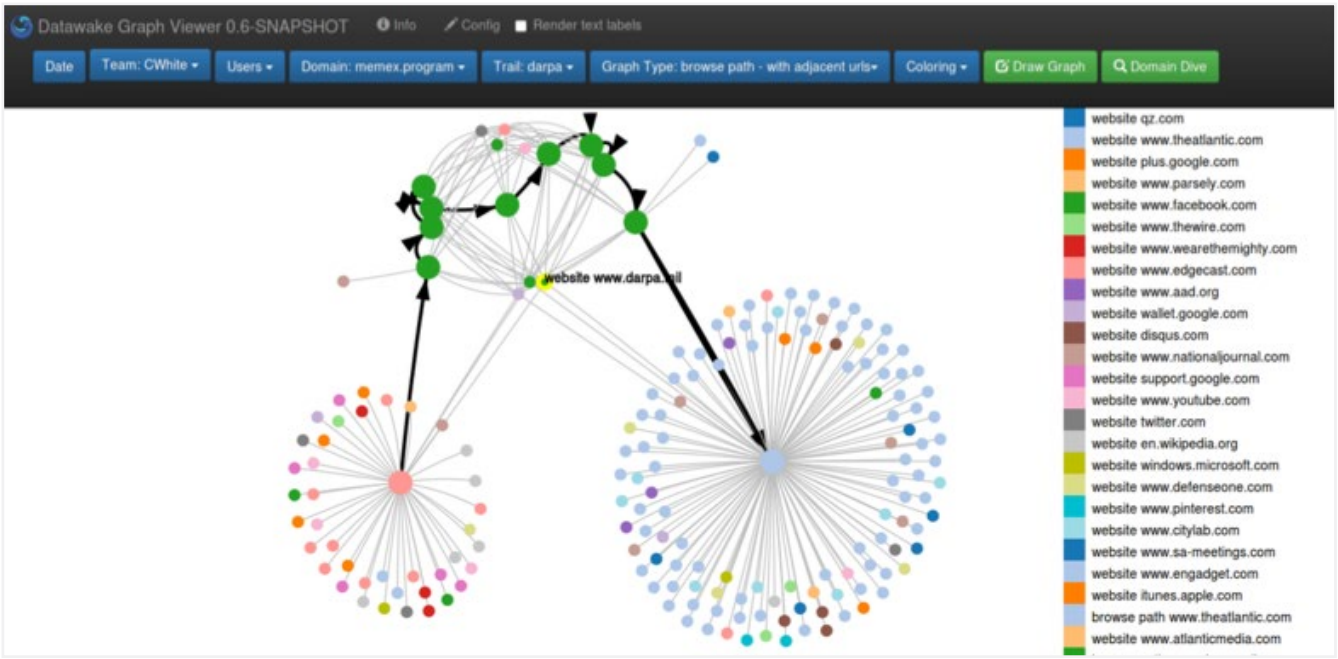
Lack of a Funded, Reinforcing Ecosystem

Proprietary vendors enjoy the advantage of a well-established, well-funded, and self-perpetuating ecosystem that tilts acquisition toward closed systems within the government marketplace. From industry connections to post-government career options, trade associations, and historical conventions, the proprietary software marketplace actively works to raise barriers to alternative acquisition and deployment models. The open source software community is not structured in this manner, and its support ecosystem typically does not focus the communities' attention beyond software development.

Recommendations

The cultural and bureaucratic hurdles to open source software are significant but ultimately surmountable. There is abundant evidence demonstrating that the increased use of open source software is possible, and even desirable, within the DoD. These examples can serve as a basis from which to develop institutional approaches to promote open source software and methods. The following recommendations offer direct and actionable ways in which the Department can rapidly increase its use of open source software to improve software outcomes and manage costs across the DoD's technology infrastructure and projects.

- The Department's senior leadership must actively embrace open source software. A lack of explicit support implies tacit support for the status quo. The Under Secretary of Defense for Acquisition, Technology, and Logistics; the DoD CIO; and service acquisitions executives and CIOs should capitalize on the opportunity provided by the White House's new open source policy, set specific goals for its integration within the Department, and interpret the national security exemption in the narrowest possible way.



DARPA's Memex program, a search tool for the dark web developed using open source methods, visualizes search results. (DARPA)

- In addition to agreeing to meet the goals set in the White House open source software policy, the DoD should actively seek to identify areas in which it can share its code with key interagency partners, including the Departments of State, Treasury, Justice, and Homeland Security.
- The Department of Defense should adopt a policy of using widely available open source software components, tools, and platforms as a default position. Proprietary software should require special approval and only be used in instances where it delivers vital functionality not provided by open source alternatives.
- The Department should establish a project or task force to develop methodologies, architectures, and repositories with which to ease the sharing and use of open source code to include the appropriate release of code via the White House's forthcoming website Code.gov, the Defense Information Systems Agency's code repository Forge.mil, GitHub, or agencies' own websites.[43] Such a project could build on the lessons learned by and methods generated from DoD projects such as Forge.mil and DARPA's Memex and xDATA. The DoD should also draw on the documented experience of sister agencies such as NASA.[44] The DoD could also provide itself and the technical public an irreplaceable benefit by creating an inspection and certification program that would help qualify and certify modules for reuse in other applications.
- The Department of Defense should collaborate with the Departments of State and Commerce to make a formal policy statement regarding the role of open source software within ITAR and export controls, specifying what is permissible under the agreements and in what ways open source licensing can speed the transfer of code to and facilitate the codevelopment of code with U.S. allies.
- The Department leadership should also examine the involvement of open source software in any future innovation and acquisition reform policies, such as in the Better Buying Power series.
- Chief technology officers and vice presidents for strategy in the defense industry should explore methods by which they can benefit from increased use of open source software. The DoD's industry partners stand to derive similar benefits as the Department itself. For businesses that develop significant quantities of custom code, open source offers important tools to maintain competitive advantage. This is particularly critical as the defense industry is forced to address long-term declines in global defense spending.

Costs of Inaction

The DoD's quest to maintain its technological superiority rests in large part on its ability to acquire, develop, deploy, and maintain cutting-edge software systems. Global C4ISR networks, precision munitions, drones, and combat aircraft are only as valuable as the data and software that power them. The martial importance of software will only increase as the Department seeks to simultaneously leverage and protect against cyber attacks, big data, artificial intelligence, and robotics in the years to come.

High quality software is also essential to the healthy function of the DoD enterprise, from accounting software to personnel records. In an era of budget uncertainty and decline, as well as ever-growing costs, the DoD needs software systems that manage its bureaucracy efficiently and enable the analyses necessary to develop elegant solutions to intractable management and fiscal challenges.

Despite these pressing needs, the DoD is manifestly not utilizing all the resources and methods available to

The DoD's quest to maintain its technological superiority rests in large part on its ability to acquire, develop, deploy, and maintain cutting-edge software systems.

ensure optimal software outcomes. Open source cannot cure all of the DoD's software ills. However, it does possess the potential to substantially improve software outcomes for the DoD in the same way it has for civilian organizations around the world, and at a lower total cost than the proprietary and closed systems the Department currently uses. Additionally, open source methods provide benefits uniquely suited to the DoD's organizational and technical needs.

In the absence of strong, sustained leadership from senior personnel in support of open source software and methods, the DoD will continue to experience unacceptable waste, pointless expense, delayed time lines, and sub-optimal outcomes from its software investments. This path guarantees a slow decline and diminution of U.S. military technical advantage – with the added possibility of periodic catastrophic failures. The U.S. military does not deserve, and cannot afford, such a future.

Endnotes

1. Department of Defense Chief Information Officer, "What is open source software (OSS)?" http://dodcio.defense.gov/OpenSourceSoftwareFAQ.aspx#Q:_What_is_open_source_software_.28OSS.29.3F.

2. Bob Work, "Deputy Secretary of Defense Speech" (CNAS Defense Forum, Washington, December 14, 2015), http://www.defense.gov/News/Speeches/Speech-View/Article/634214/cnas-defense-forum.

3. Kristina Wong, "Armed Services chair to propose new defense acquisition reforms," *The Hill*, March 15, 2016, http://thehill.com/policy/defense/273006-armed-services-committee-chair-to-propose-new-defense-acquisition-reforms.

4. Brendan McGarry, "Experts to Study F-35 Software Delays," *Defense Tech*, December 26, 2013, http://www.defensetech.org/2013/12/26/experts-to-study-f-35-software-delays/; and Chris Kanaracus, "Watchdog Agency Report Shows Beleaguered State of U.S. Military Software Projects," *PC World*, April 2, 2012, http://www.pcworld.com/article/253038/watchdog_agency_report_shows_beleaguered_state_of_us_military_software_projects.html.

5. Steve Schmidt, "Budget Cuts Even Congress Can Agree On: Software License Optimization," *Breaking Gov*, August 14, 2012, http://breakinggov.com/2012/08/14/budget-cuts-even-congress-can-agree-on-software-license-optimiz/.

6. Tony Scott, "Leveraging American Ingenuity through Reusable and Open Source Software," the White House Blog, March 10, 2016, https://www.whitehouse.gov/blog/2016/03/09/leveraging-american-ingenuity-through-reusable-and-open-source-software. White House Chief Information Officer, "Federal Source Code Policy," https://sourcecode.cio.gov/.

7. Tony Scott, "The People's Code," the White House Blog, August 8, 2016, https://www.whitehouse.gov/blog/2016/08/08/peoples-code.

8. White House Chief Information Officer, "Federal Source Code Policy," https://sourcecode.cio.gov/.

9. 44 U.S.C. § 3542, "Definitions." https://www.law.cornell.edu/uscode/text/44/3542

10. "March 2016 Web Server Survey," Netcraft, http://news.netcraft.com/archives/category/web-server-survey/.

11. "Smartphone OS Market Share, 2015 Q2," IDC Research, http://www.idc.com/prodserv/smartphone-os-market-share.jsp; and Steven J. Vaughan-Nichols, "Debunking four myths about Android, Google, and open-source," Linux and Open Source blog on *ZDNet*, February 18, 2014, http://www.zdnet.com/article/debunking-four-myths-about-android-google-and-open-source/.

12. "75% of top 10k websites served by open source software," Tech Blog on pingdom.com, May 22, 2012, http://royal.pingdom.com/2012/05/22/75-percent-top-10k-websites-served-by-open-source-software/.

13. Steven J. Vaughan-Nichols, "It's an open-source world: 78 percent of companies run open-source software," *ZDNet*, April 16, 2015, http://www.zdnet.com/article/its-an-open-source-world-78-percent-of-companies-run-open-source-software/.

14. Steve Lohr, "Some I.B.M. Software Tools to Be Put in Public Domain," *The New York Times,* November 5, 2001, http://www.nytimes.com/2001/11/05/technology/05OPEN.html.

15. Peter Levine, "Why There Will Never Be Another RedHat: The Economics Of Open Source," *TechCrunch*, February 13, 2014, http://techcrunch.com/2014/02/13/please-dont-tell-me-you-want-to-be-the-next-red-hat/.

16. Herb Caudill, "The Revolution will not be Open Source," DevResults Blog, January 20, 2016, http://blog.devresults.com/the-revolution-will-not-be-open-source/.

17. Stephen Sanzo, "Snappy Comebacks to Your Boss's Open Source Software Objections," Isovera Blog, February 2011, https://www.isovera.com/blog/snappy-comebacks-your-boss%E2%80%99s-open-source-software-objections.

18. Deborah Gage, "GitHub Raises $250 Million at $2 Billion Valuation," *The Wall Street Journal,* July 29, 2015, http://www.wsj.com/articles/github-raises-250-million-at-2-billion-valuation-1438206722.

19. Cade Metz, "Open Source Software Went Nuclear This Year," *Wired,* December 27, 2015, http://www.wired.com/2015/12/2015-the-year-that-open-source-software-went-nuclear/; Cade Metz, "Google Just Open Sourced TensorFlow, Its Artificial Intelligence Engine," *Wired*, November 9, 2015, http://www.wired.com/2015/11/google-open-sources-its-artificial-intelligence-engine/; and Elon Musk, "All Our Patent Are Belong to You," Tesla Blog, June 12, 2014, https://www.teslamotors.com/blog/all-our-patent-are-belong-you.

20. Steven Melendez, "How Facebook's Massive Open-Source Push Delivers Better Code and Better Engineers," *Fast Company,* January 26, 2015, http://www.fastcompany.com/3038842/how-facebooks-massive-open-source-push-delivers-better-code-and-better-engineers.

21. David A. Wheeler, "Open Source Software (OSS) and Total Cost of Ownership (TCO)" (Slides prepared for Government Open Source Conference 2011), http://www.dwheeler.com/essays/oss-tco-wheeler.pdf.

22. Jesús Gil Hernández, "Diseconomies of Scale in Software Development," Management and Leadership Notes Blog, February 17, 2013, http://jesusgilhernandez.com/2013/02/17/diseconomies-of-scale-in-software-development/.

23. Daniel E. Geer, Jr., "Cybersecurity and National Policy," *Harvard Law School National Security Journal*, January 10, 2011, http://harvardnsj.org/2011/01/cybersecurity-and-national-policy/.

24. Kathleen Fisher, "Using Formal Methods to Secure More Vehicles DARPA's HACMS Program," 19th Association for Computing Machinery Special Interest Group on Programming Languages International Conference, August 2014, https://www.researchgate.net/publication/269206322_Using_Formal_Methods_to_Enable_More_Secure_Vehicles_DARPA's_HACMS_Program.

25. Department of Defense, "March 2016, FY2017PB IT-1 Report (Unclassified Only)," https://snap.pae.osd.mil/snapit/BudgetDocs2017.aspx.

26. "National Security Agency," Github, https://github.com/NationalSecurityAgency/SIMP; "Open Catalog," DARPA, http://opencatalog.darpa.mil/.

27. Terry Bollinger, "Use of Free and Open-Source Software (FOSS) in the U.S. Department of Defense," MP 02 W0000101 (MITRE Corporation, January 2, 2003), 2, http://dodcio.defense.gov/Portals/0/Documents/FOSS/dodfoss_pdf.pdf.

28. Private communications, March 19, 2016.

29. Department of Defense Chief Information Officer, "Open Source Software (OSS) in the Department of Defense (DoD)," May 28, 2003, http://www.terrybollinger.com/stenbitmemo/stenbitmemo_png/index.html; and "DoD Open Source Software (OSS) FAQ," http://dodcio.defense.gov/OpenSourceSoftwareFAQ.aspx.

30. Ibid.; and "Clarifying Guidance Regarding Open Source Software (OSS)," October 16, 2009, http://dodcio.defense.gov/Portals/0/Documents/FOSS/2009OSS.pdf.

31. Cheryl Pellerin, "CIO Priorities Include Cybersecurity, Innovation, Retaining IT Workforce," DoD News, Defense Media Activity, March 23, 2016, http://www.defense.gov/News-Article-View/Article/702488/cio-priorities-include-cybersecurity-innovation-retaining-it-workforce; and "FY07 Defense Appropriation, Backgrounder: Chocola Amendment," Office of Rep. Chris Chocola (R-IN), U.S. House of Representatives (2003–06), http://m.govexec.com/pdfs/Backgrounder3.doc.

32. "Quartermaster Apps," U.S. Army, http://www.quartermaster.army.mil/quartermaster_apps.html; and "DCGSA," U.S. Army, https://dcgsa.army.mil/.

33. John Keller, "General Atomics to design Predator and Reaper UAV ground control stations with Linux processing," *Military & Aerospace Electronics*, May 25, 2011, http://www.militaryaerospace.com/articles/2011/05/general-atomics-to.html; and Noah Shachtman, "Exclusive: Computer Virus Hits U.S. Drone Fleet," *Wired,* October 7, 2011, http://www.wired.com/2011/10/virus-hits-drone-fleet/.

34. Department of Defense, "The Department of Defense Cyber Strategy," http://archive.defense.gov/home/features/2015/0415_cyber-strategy/.

35. "NSA Shares Cyber Tool on Agency's Corporate GitHub Website," National Security Agency press release, July 9, 2015, https://www.nsa.gov/public_info/press_room/2015/NSA_Shares_Cyber_Tool.shtml.

36. "The Centerpiece of 21st Century Global Security," F-35 Lightning II: Global Participation, https://www.f35.com/global.

37. Department of Defense Chief Information Officer, "DoD Information Assurance Certification and Accreditation Process (DIACAP)," Number 8510.01, ASD(NII)/DoD CIO, November 28, 2007, http://www.acqnotes.com/Attachments/DoD%20Instruction%208510.01.pdf.

38. Kristina Wong, "Armed Services chair to propose new defense acquisition reforms," *The Hill*, March 15, 2016, http://thehill.com/policy/defense/273006-armed-services-committee-chair-to-propose-new-defense-acquisition-reforms.

39. DARPA, "What is XDATA?," http://xdataonline.com/.

40. Gary M. Shiffman, "Data Analytics, Red Pills and Industrial Revolutions," WashingtonExec, March 30, 2015, http://www.washingtonexec.com/2015/03/giant-oaks-gary-shiffman-talks-data-analytics-red-pills-and-industrial-revolutions/.

41. Steven J. Vaughan-Nichols, "NSA partners with Apache to release open-source data traffic program," *ZDNet*, November 25, 2014, http://www.zdnet.com/article/nsa-partners-with-apache-to-release-open-source-data-traffic-program/.

42. Private communication.

43. Tony Scott, "The People's Code," the White House Blog, August 8, 2016, https://www.whitehouse.gov/blog/2016/08/08/peoples-code; Defense Information Systems Agency, "Forge.mil: Overview," http://www.disa.mil/enterprise-services/applications/forge-mil.

44. Chris A. Mattman et al., "Understanding Open Source Software at NASA," *IT Pro,* March/April 2012, 29–35.

About the Center for a New American Security

The mission of the Center for a New American Security (CNAS) is to develop strong, pragmatic and principled national security and defense policies. Building on the expertise and experience of its staff and advisors, CNAS engages policymakers, experts and the public with innovative, fact-based research, ideas and analysis to shape and elevate the national security debate. A key part of our mission is to inform and prepare the national security leaders of today and tomorrow.

CNAS is located in Washington, and was established in February 2007 by co-founders Kurt M. Campbell and Michèle A. Flournoy.

CNAS is a 501(c)3 tax-exempt nonprofit organization. Its research is independent and non-partisan. CNAS does not take institutional positions on policy issues. Accordingly, all views, positions, and conclusions expressed in this publication should be understood to be solely those of the authors.

1152 15th Street, NW Suite 950 Washington, DC 20005

t. 202.457.9400 | f. 202.457.9401 | info@cnas.org | cnas.org

Bold. Innovative. Bipartisan.