Value of open source projects: A case for open source cybersecurity

Michael Weiss, and Tony Bailetti
Technology Innovation Management Program
Carleton University
Ottawa, Canada
{weiss,bailetti}@sce.carleton.ca

***Abstract*—Current solutions to address cybersecurity threats take a siloed approach. Such a siloed approach favors attackers who share exploits with one another. The paper argues that cybersecurity threats are best addressed through open source projects that combine the shared expertise of security experts. To this end we develop a tool for assessing the value of open source projects and then apply it to examine how cybersecurity threats can be addressed through open source projects. This article will be relevant to organizations and government agencies who need to ensure that their networks are safe from cyberattacks, cybersecurity service providers, and to researchers interested in the intersection of open source and cybersecurity.**

Keywords—open source; business models; value creation; modularity; trust; cybersecurity

I. Introduction

Companies have become dependent on data and networks. This makes them increasingly vulnerable to cybersecurity threats. Current solutions to address cybersecurity threats take a siloed approach. This puts companies at a disadvantage over attackers: while attackers are known to share security exploits with one another, companies have been rather secretive about threats faced and their approaches to deal with them. This poses a dilemma for companies: by disclosing security attacks they risk their reputation, but can also learn about security threats and possible fixes from others; by not disclosing, it will take them longer to learn about about new threats and how to address them, giving attackers more chances to launch exploits.

In this paper, we argue that cybersecurity threats are best addressed by open source projects through which companies can share their knowledge about security vulnerabilities and expertise on fixing them. To this end, we first examine how companies derive value from open source projects in general, and create a tool that they can use to assess the value of an open source project. We then apply this tool to make our case that companies are better off addressing cybersecurity threats through open source projects rather than by pursuing individual solutions. In short, the approach of the paper is to identify how open source projects create benefits, and then to apply those lessons to deduce how cybersecurity challenges can be better addressed than in the current siloed approach. We provide evidence for the this approach by reviewing existing efforts towards an open source approach to cybersecurity.

In the sections below, we first identify the value drivers for open source projects and review how companies can engage with open source projects. We then present a tool for assessing open source projects that combines the value drivers and levels of engagement, and illustrate its use with examples from existing open source projects. Subsequently, we show that the tool allows us to argue that cybersecurity threats should be addressed through open source projects. The paper rounds up with a set of conclusions and suggestions for future research.

II. Value drivers for open source projects

Open source software has become an integral part of commercial software development [1]. In the past, open source software development was considered to be driven by volunteer effort. Yet, today, the majority of contributions to open source projects are made by developers paid to contribute. Many businesses now incorporate open source into their product or service offerings. While it is understood that companies need to engage with open source projects as part of their strategy [2], it is not understood how a company can assess whether it is deriving value from an open source project, i.e., how it can most benefit from the project.

To explore the drivers of value we draw on recent insights by Schmidt and Keil [3] that advance resource-based theory. The resource-based literature posits that superior performance over other firms is a direct result of the access to and use of superior resources [4][5][6]. However, the value of a resource is defined in terms of ex-post outcomes, i.e., after the performance of the resource is already known.

Schmidt and Keil [3] develop a theory that identifies the ex-ante conditions under which firms attribute value to a resource. They highlight the crucial difference between the ex-ante value of a resource (i.e., value before a decision to acquire or build the resource is made) and the ex-post value of a resource. Schmidt and Keil also identify four conditions that make a resource valuable to a firm ex-ante: i) the firm's ex-ante market position; ii) its ex-ante resource base, which allows for complementarities; iii) its position in inter-organizational networks; and iv) the prior knowledge and experience of its managers.

We apply the logic that Schmidt and Keil [3] used to examine the ex-ante value a firm allocates to a resource for the purpose of examining the ex-ante value a stakeholder allocates to an open source project. A stakeholder is an individual or organization that can potentially make cash or in-kind contributions to the open source project. In-kind contributions can include access to resources and people.

We postulate that, to increase its ex-ante value to a stakeholder, an open source project must increase spread, demand, complementarity, privileged information, and judgement. In earlier work [7], one of the authors (Bailetti) created a similar list of value drivers to assess the value stakeholders attribute to startups. Otherwise, the present work is independent from the earlier work. Note that we associate two value drivers with a stakeholder's market position: spread and demand. *Spread* measures by how much engaging in an open source project helps reduce the cost of product development. The cost reduction amounts to the difference between the cost of acquiring or developing proprietary software and the cost of engaging in the project. *Demand* measures how many units of a stakeholder's product are sold as a result of engaging in the open source project.

Complementarity can be expressed as the number of units sold due to the company's product complementing other products. Products complement one another when they are sold together. The existence of one complement increases the value of other complements. A company can sell complements to an open source project such as specialized hardware, software that adds new functionality, or support. The literature on modularity in open source projects [8] also informs us that modules of an open source project with different functionality are complements; the presence of each module makes the other modules more valuable to all project contributors.

Privileged information is measured as the volume, variety, velocity (i.e., timeliness), and veracity (i.e., accuracy) of privileged information that is accessible due to engaging in an open source project. Access to privileged information is provided through membership in networks [3]. Finally, *judgement* can be expressed as the number of individuals who have the requisite experience and knowledge to create value for the stakeholders attracted due to engagement in an open source project. By participating in an open source project stakeholders gain access to the experience and knowledge of other stakeholders who participate in the project.

III. Levels of engaging in open source projects

Companies that engage with open source projects either i) use open source components to develop new products, ii) build their products/services around an open source offer, iii) initiate their own open source projects, or iv) collaborate with other companies (these may include competitors) in the creation of shared open source assets. Hence, there are four levels of engaging with an open source project [9][10]:

Use. When using the outputs of an open source project, companies build their products or services by incorporating existing open source components. Their motivation for using open source is product development efficiency. Reusing code that exists shortens the time and cost of introducing their offers to the market.

Contribute. When contributing to a project, they also contribute back to the respective open source projects. By taking a more active role in these projects, companies build good will with the open source project community, but also reduce their maintenance costs down the line. The main driver is still efficiency, though.

Champion. When championing a project, a company initiates its own open source project, and aims to create an ecosystem around it. Different from the two previous stages, the company's focus is now on developing new ways of capturing value from products or services developed around the company's open source project.

Collaborate. When collaborating, a group of companies jointly develop non-strategic assets that each member of the group can incorporate into its own projects. As at the previous stage, the company's focus is now on developing new ways of capturing value. This stage also involves an ecosystem, but the assets at the core of the ecosystem are now shared among the stakeholders, unlike at the previous stage.

As Fig. 1. shows, the higher the level of engagement with an open source project, the higher the payoff.

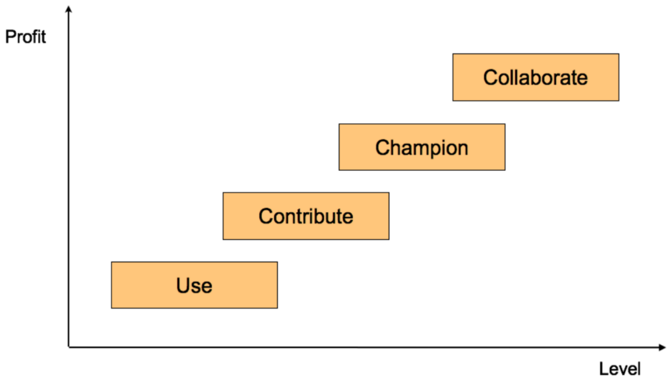

Fig. 1. Levels of engagement with open source projects.

IV. A tool for assessing open source projects

Table I is a tool in the form of a matrix that combines the five value drivers for open source projects with the four stages of engagement. Each cell of the table identifies actions that can be taken by stakeholders at a given level of engagement to drive value in a specific way. Each action takes the form: if a stakeholder does "X" at a given level of engagement, then "Y" will be the resulting value. For example, re-using open source components to develop a product reduces the cost of product development and, thus, increases spread (top left cell).

To develop the tool, we examined six open source projects that we had studied in detail in our previous research, and drew on the open source literature to complement our findings. For each project, we inferred the actions that were taken to create value from the open source project, and classified them by level of engagement and value driver. This resulted in six separate matrices, which we then collapsed into a single matrix. The matrix in Table I, thus, shows the actions that were taken in at

TABLE I. TOOL FOR ASSESSING OPEN SOURCE PROJECTS.

Level of engagement	**Value driver**				
	Increase spread	***Increase demand***	***Increase complementarity***	***Increase privileged information***	***Increase judgement***
Use	Reduce cost of development	Develop new features quickly to attract customers		Monitor technological trends	
Contribute	Reduce cost of providing standard features	Make company's product more attractive by including standard features	Create plug-ins into other products	Allocate developers to subprojects	
Champion	Attract community contributions to project	Reduce cost of acquisition for customers Allow customers to trial product	Attract third party features and services Define ownership of contributions Donate initial code for the project	Nurture the community Attract third party features and services	Access to a pool of talented developers
Collaborate	Reduce cost of creating shared assets	Create a common platform for products	Jointly create new markets	Learn from one another	Access to a diversity of skills

least some of the open source projects. Given that projects were at distinct stages of their development, and focused on different levels of engagement, not all actions would be applicable to any given project. In the next section, we provide examples for using the tool on two specific projects.

Below, we provide a rationale for the tool organized by level of engagement. Note that we could have, equivalently, organized the support for the tool by value driver. This is not a presentation of the open source engagement model itself, but a way of grouping the rationale. We noticed that while companies would focus on different stages of the engagement model, they would perform actions across the range of value drivers.

A. *Actions available at the Use level*

Reusing components from open source projects helps a company shorten the time it takes it to create the first version of a product and keep its development costs low [1].

For example, the initial version of the BigBlueButton open source web conferencing system was built by combining 14 different existing open source components [1]. This approach kept the cost of developing the initial system low and sped up the creation of the first version (*increases spread*). It helped the lead developer of BigBlueButton demonstrate the feasibility of building such a system and attract customers as well as contributors to the project (*increases demand*). However, bootstrapping also increased the complexity of the software and breadth of knowledge needed (*decreases spread*).

The impact of reusing open source components on spread and demand has been documented in Riehle's [11] study on the economics of open source. System integrators sell a solution that comprises hardware, software, and services. Money saved on software licenses by using open source components increases the revenue from services (*increases spread*). By reducing the cost of software acquisition an integrator can also bid for its services at a lower price and increase the potential number of customers (*increases demand*).

When a company uses open source components, an additional benefit is that it allows the company to monitor for technological trends (*increase privileged information*). New technologies often first emerge in experimental open source projects, and thus observing open source projects can provide a company with early information on technologies that may impact its own products' roadmap.

B. *Actions available at the Contribute level*

By contributing to an open source project, be it code, people, or money, companies achieve three goals: i) they build trust with the community, ii) they influence the direction of the project, and iii) they demonstrate their depth of competence [1][10][12]. Often these contributions are made in specific areas of expertise that most benefit the company. For example, IBM, HP, and Sun (before their acquisition by Oracle in 2010) all contributed to the development of the Mozilla Firefox browser [13]. Their contributions consisted of code that made Firefox compatible with their Unix-based workstations.

By adapting an open source project such as Firefox to make it compatible with its own operating system, a company like IBM can leverage the investments that have already been made in the open source project by others. Furthermore, by contributing any improvements made back to the open source project, the project will be better aligned with the company's needs, reducing future porting efforts [1]. These actions achieve two goals: they reduce the cost of providing a feature (e.g., web browser) that users expect (*increase spread*), and make the company's product (e.g., operating system) more attractive to potential users (*increase demand*).

The integration of a standard web browser into a vendor's proprietary operating system also illustrates complementarity. Vendors of Unix-based workstations needed the Firefox web browser to make the workstations attractive to users more familiar with PCs [13]. Blindside Networks has created a plug-in into the Learning Management System (LMS) Moodle that allows Moodle course authors to start and record BigBlueButton conferences. Third parties have subsequently also written plug-ins into other LMSs. Both actions lead to increased sales of services around hosting LMSs with BigBlueButton conferencing capability (*increase complementarity*).

C. *Actions available at the Champion level*

When a company initiates its own open source project, it does so in order to create an ecosystem of stakeholders around the project. Its focus is no longer on improving development efficiency as at the previous levels of engagement, but on discovering new ways of capturing value from products or services built around the project [9].

In order to attract outside contributors, a company needs to create and build legitimacy with a project community. Practices to build legitimacy include: i) giving to the community (e.g., free support), ii) clear assignment of ownership over contributions (property rights), iii) clear processes for making contributions, iv) transparency of decision making, and v) not treating community members as prospects [14].

External contributions are not limited to code, but include users and adopters of the open source project [15]. Users that employ the project outputs internally will increase the project's installed base. Some users will become adopters and integrate project outcomes into products and services, leading to the creation of complements (*increase complementarity*) and advocacy for the project (*increase demand*).

Finally, some adopters will become contributors and contribute advances to the project technology back to the project, and thereby reduce the cost of development (*increase spread*). Non-technology contributions such as bug reports and feature requests also come from users and adopters *(increase privileged information*).

By offering their core product at zero cost, project champions appeal to cost-sensitive customers [16]. A low-price strategy reduces the cost of software acquisition for customers and increases the number of customers the project champion can sell to (*increase demand*). Note that open source software can also be downloaded and tested before customers make an acquisition decision. This trialability decreases marketing cost (*increase spread*) and reduces demand risk [16] (*increase demand*).

By opening up a part of its product by releasing it as open source, a company can gain wider adoption of its product (even against larger incumbents) and can sell proprietary complements (e.g., support services, training, specialized hardware) to adopters [13][17]. While the open source product itself is difficult to monetize, stakeholders can charge for products or services that complement it (*increase complementarity*). The creators of the BigBlueButton web conferencing system sold add-on modules such as desktop sharing to business users. These paid features met the needs of users willing to pay to have them added to the web conferencing system.

Third parties also create products that complement the open source product. Good examples are commercial plug-ins sold for OpenOffice or Moodle (*increase complementarity*) [18]. The ecosystem of third parties includes all stakeholders that benefit from the champion's presence, e.g., providers of support services, publishers, educators, and partners who add specific capabilities to the base product [16]. For example, an ecosystem of partners has evolved around the BigBlueButton project that increase scalability of the system, integrate the system with other products (e.g., LMSs), port it to other platforms (e.g., Android), and provide internationalization.

Finally, running an open source project gives a company access to a pool of talented developers [16]. Companies like MySQL and JBoss (now RedHat) are able to recruit developers from contributors to their projects [16]. These developers are self-selecting when they join the project, and have requisite skills and experiences to advance the project [8]. Before making a hiring decision, companies already have a good record of the potential hire's contributions. By the time they are hired, these developers will also already be very familiar with the project (*increase judgement*).

D. *Actions available at the Collaborate level*

By collaborating, stakeholders can achieve outcomes that no single stakeholder could achieve on their own [19]. When stakeholders jointly develop non-strategic assets as an open source project, each stakeholder can incorporate those assets into its own projects. Good examples of open source projects that achieve this are Linux and Eclipse. Both receive a majority of their contributions from companies that otherwise compete with one another, but use the project as a common platform.

Participating stakeholders can optimize the use of their limited resources by pooling them with other stakeholders to jointly develop a common stack of open source assets that the stakeholders can then all build on to develop their individual products (*increase spread*) [17][13]. The time it takes a stakeholder to develop a product based on the open source project decreases with its level of contribution to the project (better alignment between the shared assets and its product) and with increased trust between the stakeholders (lower overhead). It also decreases with the number of stakeholders, provided that the project is suitably modularized [20].

The members of the Eclipse project develop common assets (e.g., code generation tools) that each of them requires, but that, on their own, do not create value for their customers. Developing such assets, nonetheless, requires them to dedicate resources. All members win by sharing those development costs with other members, and focus on areas of differentiation [17].

The project is a container for non-differentiating, shared assets (e.g., code, documentation) that stakeholders do not want to duplicate. This common platform allows stakeholders to focus on creating differentiating features for their products. The wider the scope of the shared assets, the wider the range of products that can be developed (*increase demand*) [17].

The Eclipse project encourages companies to incorporate shared assets into their own products [21]. Eclipse provides extension points through which developers can extend the Eclipse code base [18]. Over time, Eclipse project assets grew in diversity from core components for an integrated development environment to include components that could be applied across a range of application domains, as well as a variety of vertical solutions for specific domains [20].

TABLE II. APPLICATION OF THE TOOL TO THE ECLIPSE PROJECT.

Level of engagement	**Value driver**				
	Increase spread	***Increase demand***	***Increase complementarity***	***Increase privileged information***	***Increase judgement***
Use					
Contribute	Reduce cost of providing standard features	Make company's product more attractive by including standard features		Allocate developers to subprojects	
Champion	Attract community contributions to project	Reduce cost of acquisition for customers Allow customers to trial product	Donate initial code for the project Define ownership of contributions	Nurture the community	Access to a pool of talented developers
Collaborate	Reduce cost of creating shared assets	Create a common platform for products	Jointly create new markets	Learn from one another	Access to a diversity of skills

By adding assets to the common platform, stakeholders collaborate on the creation of new markets. The more products are being developed using the platform, the more attractive it becomes for complementors, stakeholders that develop ancillary products that expand the market for the platform, to introduce new complements (*increase complementarity*) [22].

Many of the contributors to the Eclipse project make their money from selling commercial ancillary extensions to the base platform. For example, IBM sells the WebSphere application server as a more scalable, professional version of the web application components include with Eclipse, and SAP uses Eclipse as the basis for its NetWeaver suite of development tools and as entry point to the SAP tool suite.

By collaborating with other stakeholders, stakeholders gain access to privileged information. The more stakeholders there are in a project, and the greater their diversity, the greater the volume, variety, velocity, and veracity of privileged information will be (*increase privileged information*).

The Eclipse project is structured as a collection of interconnected subprojects [20]. Each subproject has its own project champion. Stakeholders that engage in multiple subprojects can gain access to knowledge, skills, and talent through those projects at lower risk than if they conducted the research themselves. In the Apache project there is also a significant flow of information and people between projects [23].

Skills and knowledge required for complex development project (e.g., infrastructure projects such as operating systems) are rarely available within one stakeholder, and stakeholders need to collaborate to gain access to specialist skill sets. Stakeholders can get also access to a greater variety of skills, i.e., multiple ways to address problems, through collaboration (increase judgement). As noted earlier the Eclipse project grew in diversity [20], and thus the project was able to support more complex tasks that require deep domain knowledge.

V. EXAMPLES OF USING THE TOOL

We assess two examples of open source projects using the tool: one in a big company (Eclipse), the other in a small company setting (BigBlueButton). Both projects are headquartered in Ottawa, Canada. The stakeholders and the value they derive from the open source project are very different.

Table III summarizes key attributes of both projects. Setting refers to the type of stakeholders involved (big company/small company). The project initiator is the company that initiated the project. Project size refers to the number of stakeholders involved in the project (large/small), and control indicates whether project decisions are made jointly by the stakeholders, or by a single stakeholder (shared/hierarchical) [24].

Table II shows the application of the tool to the Eclipse project (www.eclipse.org). Eclipse is a project focused on building an open software development platform [25]. The Eclipse project started as an internal project at IBM that was spun out as an open source project in 2001. Initially, the Eclipse community was mainly driven by IBM. With the creation in 2004 of the Eclipse Foundation as an independent, non-profit governance body, IBM relinquished its control over the project and allowed other players, including its competitors, to become equal members of the project [26].

TABLE III. KEY ATTRIBUTES OF THE OPEN SOURCE PROJECTS.

Attribute	**Eclipse**	**BigBlueButton**
Setting	Big company	Small company
Project initiator	IBM	Blindside Networks
Project size	Large	Small
Control	Shared	Hierarchical

Table IV applies the tool to the BigBlueButton project (www.bigbluebutton.org). BigBlueButton is an open source web conferencing system. The project was initiated in 2007 at Carleton University and shortly thereafter spun out into a company, Blindside Networks, that leads the project. Blindside Networks has created an active community of contributors around the project [14]. However, unlike IBM in the case of Eclipse, it maintains control over the direction of the project and supplies the majority of contributions. External contributions increase scalability of the system, provide interfaces to other products, help port the system to other platforms, and provide internationalization.

TABLE IV. APPLICATION OF THE TOOL TO THE BIGBLUEBUTTON PROJECT.

Level of engagement	**Value driver**				
	Increase spread	***Increase demand***	***Increase complementarity***	***Increase privileged information***	***Increase judgement***
Use	Reduce cost of development	Develop new features quickly to attract customers		Monitor technological trends	
Contribute			Create plug-ins into other products (e.g., LMS)		
Champion	Attract community contributions to project	Reduce cost of acquisition for customers Allow customers to trial product	Attract third party features and services Define ownership of contributions	Nurture the community Attract third party features and services	Access to a pool of talented developers
Collaborate					

The actions in Tables III and IV were selected from the actions in Table I. Each table should be considered a “profile” of the open source project in question. It contains the actions from that were taken by the stakeholders involved in the open source project. The stakeholders' activity was derived from the literature on each project as well as from direct interaction of the researchers with key members of each of the project.

An assessment should begin by identifying the level(s) of engagement with the open source project. In the case of Eclipse, IBM and other contributors focused their engagement on the last three levels, whereas in the case of BigBlueButton, Blindside Networks focused on the first three. Correspondingly, the emphasis in the Eclipse project was on actions related to creating a common platform of shared assets, while the BigBlueButton project emphasized community creation and attracting complementors.

The actions are, by necessity, only templates. As appropriate, the actions should be made more specific when assessing an open source project. This is what we have done when assessing the potential of open source projects for addressing cybersecurity threats (see Section VI).

VI. TOWARDS AN OPEN SOURCE APPROACH TO CYBERSECURITY

In this section, we apply the tool for assessing open source projects developed in this paper to examine how cybersecurity threats can be addressed through open source projects.

As stated earlier, the siloed approach to addressing cybersecurity threats puts companies at a disadvantage vis à vis attackers. While attackers are known to share security exploits with one another, companies have been rather secretive about threats faced and their approaches to deal with them. Known modes of collaboration among attackers include [27]: crimeware-as-a-service (e.g., rental of malware), pay-per-install or pay-per-infection, crimeware toolkits, brokers that act as a trusted intermediary between sellers and buyers of exploits, and data suppliers (e.g., of password lists).

Some have called for companies to collaborate in dealing with cybersecurity threats [28][29][30]. Ackerman [28] argues that every aspect of business has become dependent on data and networks, and thus cybersecurity. He calls for companies to share the risk in building a cybersecurity defence by taking a collaborative approach, in which companies share expertise and threat intelligence. His rationale is that addressing cybersecurity challenges jointly, rather than operating in “silos”, will allow companies to build more powerful defence mechanisms against attackers.

Ackerman [28] documents different means for companies to collaborate: special interest collaborations, cybersecurity bounties, and shared threat intelligence. These efforts harness the shared expertise of a community of stakeholders, and allow stakeholders, who individually do not have the necessary skills and financial resources, to detect and address critical vulnerabilities to improve their cybersecurity defence.

Shiffman and Gupta [29] argue for the creation of a cybercommons where stakeholders organically establish rules that regulates behavior in cyberspace via a bottom-up or collective approach, quite unlike a centralized approach where rules are imposed in a top-down manner. In a cybercommons collectives of security experts organizes themselves to address security threats. It falls to institutions (known as keystones in an ecosystem [19]) to provide the space where security experts and other stakeholders can build trusted relationships and interact.

The Conficker Working Group is an example of the bottom up approach to cybersecurity [29]. Conficker was a computer virus that spread rapidly and was particularly difficult to trace. A group of computer security experts with different affiliations formed a voluntary community, the Conficker Working Group, in order to pool threat intelligence and expertise [28]. The effort was also noteworthy because of its collaboration between private and public organizations in different regions of the world.

In his review of current internet security policy, Schmidt [30] makes a case for “open security” to be provided by non-state, global, user-serving, more accountable actors instead of traditional public-private partnerships and closed-door operational security communities (e.g., CERTs). Among the goals he identifies for open security are the need for cultural diversity, further globalization, intrinsic motivation of contributors, a flat governance structure, and more transparency and openness.

Swire [31] examines under which conditions disclosure of security threats is socially optimal, when taking into account

TABLE V. OPPORTUNITIES FOR USING OPEN SOURCE PROJECTS TO ADDRESS CYBERSECURITY THREATS

Level of engagement	**Value driver**				
	Increase spread	***Increase demand***	***Increase complementarity***	***Increase privileged information***	***Increase judgement***
Use	Reduce cost of development	Increase security for their products and services		Monitor technological trends	
Contribute			Share security expertise		
Champion			Create a platform for sharing threat intelligence and security expertise Define ownership of contributions	Nurture the community Attract third party features and services	Access to a pool of talented developers
Collaborate	Reduce cost of creating shared assets			Share threat intelligence	Access to a diversity of security expertise

the benefits and costs to both attackers and defenders. He argues that revealing the details of implementing a system, as in the open source approach, will not help attackers in a world where attackers share information and exploits are quickly learned by others. Rather, disclosing a threat will allow others to improve the design of the system's defences. Disclosure will also allow others to patch their systems or otherwise protect themselves against the threat.

Table V identifies opportunities for using open source projects to address cybersecurity threats. Companies benefit from engaging with open source projects (as users, contributors, champions, or collaborators) by reducing the cost of developing solutions to security threats, increasing security for their products and services, and having access to threat intelligence and security expertise.

VII. CONCLUSION

In this paper we developed a tool for assessing the value of open source projects. The purpose of the tool is to help companies increase the ex-ante value they receive from engaging with open source projects. The foundation for the tool was provided by a recent theoretical advance in resource-based theory [3], our research on open source engagement [1][9][20] and business ecosystems [19][21], and our experience gained from working closely with two open source projects.

The tool combines five factors that make an open source project valuable by applying the logic of [3] with four levels of engagement in open source projects identified in our earlier research [9]. The tool can be used to describe actions to be taken by stakeholders at a given level of engagement to drive value in a specific way. We applied it to examine how cybersecurity threats can be addressed through open source projects. Our interest in the application to cybersecurity stems from the fact that open source approaches have not yet been widely applied in cybersecurity, even though there have been several calls for such an approach. We are also currently involved in a major research project on cybersecurity [32].

Future research should conduct an empirical analysis of the actions of stakeholders engaging with open source projects by applying the tool to a larger sample of open source projects. Another area for future work is to examine current projects in the still nascent field of open source cybersecurity through the lens of the tool. One goal of this research would be to identify additional ways in which stakeholders can benefit from an open source approach to cybersecurity.

REFERENCES

[1] M. Weiss, “Profiting from open source,” European Conference on Pattern Languages of Programs, ACM, article no. 5, 2010.

[2] T. Bailetti, “How open source strengthens business models,” Technology Innovation Management Review, February, 2009, http://timreview.ca/article/226.

[3] J. Schmidt, and T. Keil, “What makes a resource valuable? Identifying the drivers of firm-idiosyncratic resource value,” Academy of Management Review, vol. 38(2), pp. 206-228, 2013.

[4] J.B. Barney, “Firm resources and sustained competitive advantage,” Journal of Management, vol. 7, pp. 99-120, 1991.

[5] M.A. Peteraf, “The cornerstone of competitive advantage: A resource-based view,” Strategic Management Journal, vol. 14, pp. 179-191, 1993.

[6] D.G. Sirmon, M.A. Hitt, and R.D. Ireland, “Managing firm resources in dynamic environments to create value: Looking inside the black box,” Academy of Management Review, vol. 32(1), pp. 273-292, 2007.

[7] T. Bailetti, and E. Zeijdemans, “Cybersecurity startups: The importance of rapid and early globalization,” Technology Innovation Management Review, November, 2014, http://timreview.ca/article/845.

[8] C. Baldwin, and K. Clark, “The architecture of participation: Does code architecture mitigate free writing in the open-source development model?,” Management Science, vol. 52(7), pp. 1116-1127, 2006.

[9] P. Carbone, “Competitive open source,” Technology Innovation Management Review, July, 2007, http://timreview.ca/article/93.

[10] M. Weiss, “The business of open source,” talk, http://www.slideshare.net/mrw/business-of-open-source-29179493, 2013.

[11] D. Riehle, “The economic motivation of open source software: Stakeholder perspectives,” Computer, pp. 25-31, April, 2007.

[12] L. Dahlander, “A man on the inside: Unlocking communities as complementary assets,” Research Policy, vol. 35(8), pp. 1243-1259, 2006.

[13] J. West, and S. Gallagher, “Challenges of open innovation: the paradox of firm investment in open-source software,” R&D Management, vol. 36(3), pp. 319-331, 2006.

[14] F. Dixon, “Lessons from an open source business,” Technology Innovation Management Review, May, 2011, http://timreview.ca/article/441.

[15] I. Skerrett, “Best practices in multi-vendor open source communities,” echnology Innovation Management Review, Januar, 2011, http://timreview.ca/article/409.

[16] R. Watson, M.C. Boudreau, P. York, M. Greiner, and D. Wynn, “The business of open source,” Communications of the ACM, vol. 51(4), pp. 41-46, 2008.

[17] M. Weiss, “Profiting even more from open source,” European Conference on Pattern Languages of Programs, ACM, article no. 1, 2011.

[18] Noori, N., and Weiss, M., “Managing the quality of platform complements: The case of extensions in open source platforms,” ISPIM Conferences, issue 23, 2012.

[19] T. Bailetti, “Blueprint and approach to share revenue in small technology companies,” Technology Innovation Management Review, June, 2010, http://timreview.ca/article/355.

[20] M. Weiss, “Economics of collectives,” International Software Product Line Conference, volume 2, article no. 39, 2012.

[21] S. Muegge, “Business ecosystems as institutions of participation: A systems-perspective on community-developed platforms,” Technology Innovation Management Review, November, 2011, http://timreview.ca/article/495.

[22] S. Muegge, “Platforms, communities, and business ecosystems: Lessons learned about technology entrepreneurship in an interconnected world,” Technology Innovation Management Review, January, 2013, http://timreview.ca/article/655.

[23] Weiss, M., Moroiu, G., and Zhao, P., “Evolution of open source communities. International Conference on Open Source Systems,” pp. 21-32, Springer/IFIP (2006)

[24] Weiss, M., “Control and diversity in company-led open source projects,” Technology Innovation Management Review, http://timreview.ca/article/436, April (2011)

[25] Smith, D., and Milinkovich, M., “Eclipse: A premier open source community,” Technology Innovation Management Review, http://timreview.ca/article/94, July, 2007.

[26] S. Spaeth, M. Stuermer, and G. v. Krogh, “Enabling knowledge creation through outsiders: Towards a push model of open innovation,” International Journal of Technology Management, vol. 52(3/4), pp. 411-431, 2010.

[27] M. Gad, “Crimeware marketplaces and their facilitating technologies,” Technology Innovation Management Review, November, 2014, http://timreview.ca/article/847.

[28] R. Ackerman, “Crowdsourcing & cybersecurity: who do you trust? Dark Reading,” 2014, http://www.darkreading.com/analytics/crowdsourcing-and-cyber-security-who-do-you-trust/a/d-id/1278747.

[29] G. Shiffman, and R. Gupta, “Crowdsourcing cyber security: A property rights view of exclusion and theft on the information commons,” International Journal Of The Commons, vol. 7(1), pp. 92-112, 2013, http://www.thecommonsjournal.org.

[30] A. Schmidt, “Open security. Contributions of networked approaches to the challenge of democratic internet security governance,” in: R. Radu, J.M. Chenou, and R. Weber (eds.), The Evolution of Global Internet Governance, Springer, 2014.

[31] P. Swire, “A Theory of disclosure for security and competitive reasons: Open source, proprietary software, and government systems,” Houston Law Review, vol. 42(5), pp. 1333-1380, 2006.

[32] T. Bailetti, D. Craigen, D. Hudson, R. Levesque, and S. McKeen, “Developing an innovation engine to make Canada a global leader in cybersecurity,” Technology Innovation Management Review, August 2013, http://timreview.ca/article/711.