Technology Innovation Management Review

February 2016
Volume 6 Issue 2

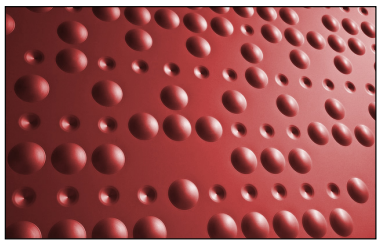

Image credit: Tom Bech (CC-BY)

Cybersecurity

Welcome to the February issue of the *Technology Innovation Management Review*. We welcome your comments on the articles in this issue as well as suggestions for future article topics and issue themes.

Carleton UNIVERSITY

www.timreview.ca

Technology Innovation Management Review

February 2016
Volume 6 Issue 2

Publisher

The *Technology Innovation Management Review* is a monthly publication of the Talent First Network.

ISSN

1927-0321

www.timreview.ca

Overview

The *Technology Innovation Management Review* (TIM Review) provides insights about the issues and emerging trends relevant to launching and growing technology businesses. The TIM Review focuses on the theories, strategies, and tools that help small and large technology companies succeed.

Our readers are looking for practical ideas they can apply within their own organizations. The TIM Review brings together diverse viewpoints – from academics, entrepreneurs, companies of all sizes, the public sector, the community sector, and others – to bridge the gap between theory and practice. In particular, we focus on the topics of technology and global entrepreneurship in small and large companies.

We welcome input from readers into upcoming themes. Please visit timreview.ca to suggest themes and nominate authors and guest editors.

Contribute

Contribute to the TIM Review in the following ways:

- Read and comment on articles.
- Review the upcoming themes and tell us what topics you would like to see covered.
- Write an article for a future issue; see the author guidelines and editorial process for details.
- Recommend colleagues as authors or guest editors.
- Give feedback on the website or any other aspect of this publication.
- Sponsor or advertise in the TIM Review.
- Tell a friend or colleague about the TIM Review.

Please contact the Editor if you have any questions or comments: timreview.ca/contact

About TIM

The TIM Review has international contributors and readers, and it is published in association with the Technology Innovation Management program (TIM; timprogram.ca), an international graduate program at Carleton University in Ottawa, Canada.

Editorial: Cybersecurity

Chris McPhee, Editor-in-Chief
Dan Craigen, Guest Editor

From the Editor-in-Chief

Welcome to the February 2016 issue of the *Technology Innovation Management Review*. This month's editorial theme is **Cybersecurity**, and I am pleased to welcome guest editor **Dan Craige**n, Science Advisor at the Communications Security Establishment and Visiting Scholar at Carleton's Technology Innovation Management Program in Ottawa, Canada.

In addition to four articles on cybersecurity, this issue also includes a summary of the first TIM Lecture of 2016, which was held in celebration of the TIM Review's 100th issue (timreview.ca/issue/2015/november). **Peter Carbone** and **Sean Silcoff** shared lessons from studying key factors that have led to success and failure in technology businesses. This event also marked the launch of a new book comprising of the journal's 15 most popular articles as the latest installment in the Best of TIM Review book series (timbooks.ca).

We hope you enjoy this issue of the TIM Review and will share your comments online. For upcoming issues, we welcome your submissions of articles on technology entrepreneurship, innovation management, and other topics relevant to launching and growing technology companies and solving practical problems in emerging domains. Please contact us (timreview.ca/contact) with potential article topics and submissions.

Chris McPhee
Editor-in-Chief

From the Guest Editor

It is my pleasure to be the guest editor for the February 2016 issue of the TIM Review. This is the seventh issue on the theme of **Cybersecurity** published in the TIM Review since July 2013.

On reviewing the seven issues (timreview.ca/issue-archive), one finds a breadth of cybersecurity research drawing, at times, from atypical sources. Examples include: i) the application of design science, ii) the utilization of club theory, iii) a description of crimeware marketplaces and their facilitating technologies, and iv) an investigation of effective digital channel marketing for cybersecurity solutions. It is through these kinds of multidisciplinary thinking that novel insights potentially arise. This issue continues the trend of multidisciplinary thinking through, for example, the proposal to use crowdsourcing for (cybersecurity) literature reviews and the use of a general model to understand the modes suppliers use to deliver of goods and services, which is applied to the context of malware.

In the first article, **Michael Weiss** discusses the application of crowdsourcing to literature reviews in new domains. Informed by recent literature reviews in cybersecurity and a discussion on the goals and types of literature reviews, Weiss develops design principles and a conceptual model for a platform for crowdsourcing literature reviews. A prototype of the platform is currently being implemented.

Next, **Tony Bailetti**, **Mahmoud Gad**, and **Ahmed Shah** introduce and define the concept of "intrusion learning". Intrusion learning is an emergent discipline that draws from machine learning, intrusion detection, and streaming network data. The expectation is that intrusion learning will significantly improve enterprise perimeter protection.

In the third article, **Tony Bailetti** and **Mahmoud Gad** apply a formal model to analyze the modes by which malware suppliers provide goods and services to their clients. A formal approach to characterizing the modes in which malware suppliers function will enhance capacity to mitigate cyberattacks.

Editorial: Cybersecurity

Chris McPhee and Dan Craigen

Finally, **Ahmed Shah**, **Selman Selman**, and **Ibrahim Abualhaol** examine open source cybersecurity packages to determine whether there are license compliance issues that could potentially result in expensive remediation costs, damage to a company's reputation, and costly legal fees. Of the 343 open source cybersecurity tools that they examined, four were found to include restrictive licenses.

The authors of the four articles in this issue are all associated with Carleton University's Technology Innovation Management program or the VENUS Cybersecurity Corporation:

- **Ibrahim Abualhaol** is a graduate of the Technology Innovation Management program.
- **Tony Bailetti** is an Associate Professor in the Sprott School of Business and the Department of Systems and Computer Engineering at Carleton University, and he is the Director of the Technology Innovation Management program.
- **Mahmoud Gad** is a research associate with the VENUS Cybersecurity Corporation.
- **Selman Selman** is a software engineer with the Software Integrity Group at Synopsys and a graduate student in the Technology Innovation Management Program.
- **Ahmed Shah** is a graduate student in the Technology Innovation Management Program.
- **Michael Weiss** is an Associate Professor in the Department of Systems and Computer Engineering at Carleton University, and a faculty member in the Technology Innovation Management Program.

I hope that you enjoy this seventh issue on the theme of Cybersecurity.

Dan Craigen
Guest Editor

About the Editors

Chris McPhee is Editor-in-Chief of the *Technology Innovation Management Review*. He holds an MASc degree in Technology Innovation Management from Carleton University in Ottawa, Canada, and BScH and MSc degrees in Biology from Queen's University in Kingston, Canada. Chris has over 15 years of management, design, and content-development experience in Canada and Scotland, primarily in the science, health, and education sectors. As an advisor and editor, he helps entrepreneurs, executives, and researchers develop and express their ideas.

Dan Craigen is a Science Advisor at the Communications Security Establishment in Canada and a Visiting Scholar in the Technology Innovation Management Program at Carleton University in Ottawa, Canada. Previously, he was President of ORA Canada, a company that focused on High Assurance/Formal Methods and distributed its technology to over 60 countries. His research interests include formal methods, the science of cybersecurity, and technology transfer. He was the chair of two NATO research task groups pertaining to validation, verification, and certification of embedded systems and high-assurance technologies. He received his BScH and MSc degrees in Mathematics from Carleton University.

Citation: McPhee, C., & Craigen, D. 2016. Editorial: Cybersecurity. *Technology Innovation Management Review*, 6(2) 3–4. http://timreview.ca/article/962

Keywords: cybersecurity, new domains, literature reviews, intrusion, machine learning, malware, multisided platforms, open source, licensing

Crowdsourcing Literature Reviews in New Domains

Michael Weiss

> *"This is a needle in a haystack problem where the appearance of the needle is unknown."*
>
> Lin et al. (2014)

Conducting a literature review in new domains presents unique challenges. The literature in a new domain is typically broad, fragmented, and growing quickly. Because little is known about the new domain, the literature review cannot be guided by established classifications of knowledge, unlike in an existing domain. Rather, it will be driven by evidence that challenges and extends existing knowledge. In a way, exploring a new domain means looking for anomalies in the evidence that cannot be explained by what is already known. This article summarizes lessons from conducting two literature reviews in new domains in the area of cybersecurity. It then presents a design for using leader-driven crowdsourcing to collect evidence and synthesize it into insights in a new domain. The article will be relevant to those who are exploring a new domain, in particular students, researchers, and members of R&D projects in industry.

Introduction

A standard approach to exploring a domain is to conduct a literature review. However, conducting a literature review in a *new* domain presents unique challenges. Whereas in an existing domain, researchers can use established classifications of knowledge to guide their search for and interpretation of the literature, this is not the case for a new domain that lacks such classifications. In a new domain, the literature is typically broad, fragmented, and, at the same time, growing quickly. The task of the researcher is to make sense of evidence when it does not fit existing models and classifications. Encountering such evidence forces them to extend existing knowledge.

This article first examines the characteristics of new domains and summarizes lessons from conducting two literature reviews in new domains. It then reviews the goals and types of literature reviews and the typical structure of a systematic narrative literature review. Third, it introduces crowdsourcing as a technique for leveraging groups of people to solve complex tasks and examines the problems crowdsourcing can solve. The article then presents a design for crowdsourcing the creation of literature reviews to collect evidence and synthesize it into insights in a new domain. The article closes with the identification of challenges and open questions when using this new approach.

Exploring New Domains

Exploring a new domain can be conceptualized as looking for anomalies in the evidence that cannot be explained by what is already known, and subsequently building models and classifications that incorporate this evidence. A particular challenge in exploring a new domain is that the very criteria for searching the domain are co-evolving with our understanding of the domain. At the outset of the literature review, there are few established criteria for what the researchers should be looking for, something that Lin and colleagues (2014), in their study on crowdsourcing the search for Genghis Khan's tomb, refer to as a "needle in a haystack problem where the appearance of the needle is unknown".

Crowdsourcing Literature Reviews in New Domains

Michael Weiss

Searching for unknowns

Very little is known about what the key concepts in the new domain are. Thus, researchers should not limit their search criteria to what can be ''expected'' based on existing literature. As observed by Attenberg, Ipeirotis, and Provost (2011), organizations make decisions based on explicit or implicit models of the world. While it is important to understand where these models have limitations and can be improved, it is often not clear when these models are limited. In other words, we often don't know what we don't know.

From requirements engineering, we also know that ignorance of a domain often has advantages (Berry, 1995). It allows a requirements analyst to uncover unstated assumptions that domain experts have come to accept. Experts have tacit knowledge of a domain (aspects of the domain they take for granted), whereas an ignorant "newbie" in the domain would have to think about those aspects explicitly and evaluate them from first principles (Mehrotra & Berry, 2012). We consider this outsider's perspective as the "newbie's advantage".

Attenberg and colleagues (2011) also recognize the advantage of a non-expert's perspective. They found that non-experts can easily find holes in decision models that pass "standard" tests used by experts. These holes in an organization's decision model correspond to situations where the model is confident but wrong (these are "unknown unknowns"), not where the model is uncertain ("known unknowns"). From these observations, we conclude that, in a new domain, researchers should especially be looking for areas in the existing knowledge that are supposedly firmly established. This is where the biggest blind spots may lie.

Lessons from two literature reviews in new domains

The author had an opportunity to observe two teams conducting literature reviews in new domains within the area of cybersecurity (see the Acknowledgements). The teams consisted of experienced researchers and graduate students and early-career researchers. All team members had prior experience writing traditional literature reviews. The key observations were:

1. *Fragmentation and size of domain:* There were not yet established classifications of the knowledge in the new domains and the knowledge appeared fragmented. This observation was more apparent in one of the reviews, which lacked a reference point for starting the literature search.
2. *Evolving search criteria:* Questions drive the search for evidence and the search criteria evolve with the understanding of the domain. Competing interpretations require adjustments to the search criteria.
3. *Output of literature review:* The intent of the literature review is to obtain a sense of the future evolution of the domain, and to identify gaps and challenges. While, in some sense, every literature reviews strives to achieve those goals, a literature review in a new domain will put more emphasis on these aspects. Our understanding of a new domain starts with gaps in and challenges to the existing literature.
4. *Grounded in examples:* The review is grounded in examples of the phenomenon investigated.
5. *Non-traditional sources of literature:* Because the domain is still evolving, other sources than traditional conference and journal papers need to be considered (e.g., online presentations and news articles).
6. *Diversity:* In these two cases, the team consisted of generalists and specialists. The generalists in the team had a broad background in technology and innovation, whereas the specialists had expertise in cybersecurity. However, none of them had specific expertise in the emerging domains.
7. *Modularity:* The search and interpretation of literature was chunked into independent pieces. This observation is more applicable to one of the reviews, where scoping the domain into subdomains helped focus the review process.
8. *Leader-driven scoping and synthesis:* Questions (scoping) and synthesis of the answers were driven by one individual (an experienced researcher), who took a lead role in the literature review process.

Table 1 summarizes the evidence for these observations, which the author solicited by email from the team members. The team members were presented with an initial version of the eight observations above and asked to comment on them. The quotations are provided as they appeared in the emails, except for correcting obvious spelling or grammatical errors.

Crowdsourcing Literature Reviews in New Domains

Michael Weiss

Table 1. Evidence collected from the authors of two literature reviews in new domains

Observation	Evidence
1. Fragmentation and size of domain	"With [Review 2] we had Machine Learning and its structures to start with. So, that classification exists. But, especially with the new evolving techniques, papers were rare and scattered."
	"[Review 1] didn't really have a starting structure at all, and we had to search around for the meaning of code reuse."
2. Evolving search criteria	"[We] used survey papers, where possible, to start the exploration and capture a broad context that would be refined."
	"[We] started by looking for recent papers (post 2010) based on evolving sets of keywords as we explored the domain."
	"I'm trying to show keyword search evolution in three phases. I would consider the first process was an initial validation of keywords to use for search, the second phase was a top-down approach for finding literature, and the third phase was a bottom-up approach to search.
3. Output of literature review	"Overall, the intent of the article is to contribute an understanding of the recent literature and a sense of future directions."
	"Certainly trying to capture trends, but also to identify gaps, challenges, etc."
4. Grounded in examples	"The review is grounded in examples of the phenomenon investigated. [Review 1] collects examples of code reuse covered by the media."
	"[We] extracted snippets from papers making points that we deemed of interest as pertaining to the focus questions from the client [funding the literature review]. [It] helped to ground the review as we moved forward in a domain [where] we were not experts."
5. Non-traditional sources of literature	"For [Review 1], we had to search outside Google Scholar for definitions and tutorials including YouTube videos."
6. Diversity	"I agree that team diversity is important – especially in a novel domain. In this case, I think it could be argued that we had no specialists on the team. I guess it depends upon the abstraction. Am I a specialist because I have insights into cybersecurity?"
7. Modularity	"True for [Review 2]; slightly less true for [Review 1]. [Review 2] resulted in five modules being written on specific subject matters (that, in a way, reflected the overall categorizations within Machine Learning)."
	"[Researcher B] modularized/scoped the research into subdomain topics that I thought included foundational elements of machine learning (i.e., feature extraction, clustering, datasets)."
8. Leader-driven scoping and synthesis	"I think in spirit this is true. But, in a way, it was the contract that set up the questions/direction. [Researcher A], however, certainly was key in determining the process used to perform the literature review."
	"In both [Review 1] and [Review 2], the project leader had to define the depth of the search as well as the level of details in the synthesis."
	"I concur; ultimately someone had to make a decision on scope."

Crowdsourcing Literature Reviews in New Domains

Michael Weiss

Goals and Types of Literature Reviews

A literature review aims to summarize the current knowledge on a given topic based on previously published research. The authors of a literature review search through the literature, retrieve sources of information, and synthesize the findings of those sources into one paper (Green et al., 2006). We can classify literature reviews in terms of their goals and the ways in which the literature review is conducted. Baumeister and Leary (1997) identify five possible goals of a literature review. Starting with the most ambitious goal, these are: developing theory, evaluating theory, surveying the state of knowledge on a particular topic, identifying a gap or a problem, and, in some cases, providing a historical account of the development of theory and research on a particular topic.

Green et al. (2006) differentiate three broad categories of literature reviews:

1. Narrative literature reviews synthesize the findings of literature retrieved from searches of databases, manual searches, and authoritative texts. They are helpful when presenting a broad perspective on a topic. However, they are usually less systematic and comprehensive than other types of literature reviews and may be biased to one researcher's perspective. Editorials, commentaries, and overview articles are all examples of narrative literature reviews.

2. Qualitative systematic literature reviews are based on a detailed search of the literature. They are driven by a focused question or purpose. A systematic literature review aims to decrease the amount of bias that can occur when evidence is extracted from the literature by establishing systematic criteria for selecting literature to include in the survey and including multiple authors in the review. Results of the review are typically compiled in evidence tables.

3. Quantitative systematic literature reviews synthesize the results of the reviewed literature in a statistical manner. A quantitative literature review is also known as a meta-analysis.

It is also possible to create a taxonomy of literature reviews by combining the goals and types of literature reviews (Pare et al., 2015). Other authors such as Grant & Booth (2009) have created more detailed classifications of literature reviews. The literature reviews conducted by the two teams above can best be characterized as a systematic narrative literature review. They are more systematic than a narrative literature review, but do not meet all the formal requirements of a qualitative systematic literature review. This type of literature review is the focus of our paper.

Structure of a Systematic Narrative Literature Review

Green, Johnson, and Adams (2006) describes a (systematic) narrative literature review as a "best-evidence synthesis". A best-evidence synthesis contains the following elements:

1. *Focus:* The authors should state the purpose or focus of the literature review.

2. *Relevance:* The authors also need to make a case for the relevance of the review.

3. *Glossary:* The literature review should define any unusual terminology.

4. *Sources of information:* The authors of the literature review need to report on the electronic databases searched and the keywords used to search for papers.

5. *Search terms:* To limit the number of papers that need to reviewed, the authors should turn the main concepts of the domain under exploration into search terms.

6. *Selection criteria:* The literature review should describe on what grounds papers were included or excluded. Such criteria help avoid bias in the selection of the papers.

7. *Synthesis:* The information obtained from the literature should be organized into common themes or streams. Tables are a good way of categorizing the evidence collected. A goal of the synthesis is to identify agreements, disagreements, and gaps in the literature.

8. *Limitations:* The authors should identify weak points of the review and areas for future work.

9. *Conclusion:* The conclusion should relate back to the purpose and summarize the major findings of the literature review and identify the contributions to knowledge made.

Crowdsourcing Literature Reviews in New Domains

Michael Weiss

Crowdsourcing

Crowdsourcing is a technique for leveraging a group of people (the crowd) to solve complex tasks. In crowdsourcing, there are two types of users: requesters and members of the crowd (Bigham et al., 2015). Requesters are the people or organizations who define a problem or task, and aggregate the partial solutions produced by the crowd. Crowd members are people who contribute. Crowdsourcing is a special type of co-creation, a practice where developers and stakeholders collaborate to create a product or service (Pater, 2009). However, unlike Pater (2009) we do not limit crowdsourcing to a scenario where anyone can join the crowd, but also include the case where crowd members are selected based on participation criteria, such as, for their expertise or collaboration history.

Types of crowdsourcing
Crowdsourcing systems differ in terms of the incentives of requesters and crowd members, the complexity of the tasks, the amount of time crowd members spend on tasks, the level of collaboration between crowd members, and in terms of whether the work is done as part of "standard" work or not. Bigham, Bernstein, and Adar, (2015) distinguish between three types of crowdsourcing: directed crowdsourcing, collaborative crowdsourcing, and passive crowdsourcing.

1. In directed crowdsourcing, a single requester recruits the members of the crowd to pursue a specific goal. In this type of crowdsourcing, the members of the crowd generally act independently. A good example of directed crowdsourcing is Amazon's Mechanical Turk platform (mturk.com), in which workers get paid for performing specified tasks for the requester. In directed crowdsourcing, large tasks are often decomposed into so-called microtasks.

2. In collaborative crowdsourcing, the crowd self-determines their organization and work. In this type of crowdsourcing, members of the crowd are usually intrinsically motivated to participate, that is, they share in interest in accomplishing a joint task such as the creation of an online encyclopedia as in the case of Wikipedia, or identifying features on satellite images such as shapes that may indicate the location of a tomb (Lin et al., 2014).

3. In passive crowdsourcing, the crowd produces a useful outcome as part of their regular behaviour. Instead of directing the activity of the crowd, the requester is simply collecting traces of the crowd's behaviour and drawing inferences from them. An example of passive crowdsourcing is tracking messages on Twitter to predict a political outcome (iHub Research, 2013).

An interesting hybrid between directed crowdsourcing and collaborative crowdsourcing is leader-driven crowdsourcing. In this type of crowdsourcing, a leader maintains a high-level vision of the task and directs other crowd members (contributors) to make specific contributions towards this task. An example of leader-driven crowdsourcing is the collaborative writing system called Ensemble (Kim et al., 2014). We will build on the concept of leader-driven crowdsourcing in the proposed design below.

Benefits of crowdsourcing
Crowdsourcing is beneficial for a number of reasons, including:

1. *Time:* By distributing a task across a large group, crowdsourcing can reduce the time it takes to complete the task, given a clear division of the task into subtasks (Brown et al., 2014).

2. *Validation criteria:* Lacking a pre-existing reference for what constitutes an anomaly in the new domain, consensus can be used as a training mechanism for the crowd (Lin et al., 2014).

3. *Diversity:* A crowd can provide access to a diversity of perspectives (André et al., 2014).

4. *Domain knowledge required:* When appropriately structured, complex problems can be solved by crowds with little to no pre-existing domain knowledge (Bigham et al., 2015).

5. *Scale:* When a task is distributed among the members of a crowd, much larger tasks can be addressed such as large-scale surveys of datasets (Lin et al., 2014).

Other work on crowdsourcing literature reviews
The application of crowdsourcing to exploring new domains has not been widely studied yet. Most applications are in the medical domain, for example finding papers that mention certain diseases or drugs (Good et al., 2015) or searching for treatments (Elliot et al., 2014), and in education, for example learning new concepts (Luther et al., 2015). Although most of the early work on crowdsourcing has focused on datasets in domains that

Crowdsourcing Literature Reviews in New Domains

Michael Weiss

most users are familiar with, such as images or travel advice, recent research has developed techniques that can deal with more complex qualitative datasets in unfamiliar domains, such as synthesizing textual data that requires domain-specific knowledge (André et al., 2014).

A search on Google Scholar for combinations of the keywords "crowdsourcing" and "literature review", "new domains", or "unfamiliar domains" only found two examples of crowdsourcing used to conduct a literature review, both in the medical domain. In the first, Brown and Allison (2014) describe a process for evaluating the literature that involves decomposing a research question of interest into microtasks that can be distributed to members of the crowd. In the second, Elliot, Thomas, and Owens (2014) describe an ongoing initiative for crowdsourced screening of citations (Embase, 2016).

One lesson from Brown and Allison (2014) is that quality checks are essential not only to guarantee the validity of the results. Quality checks are also required to demonstrate the competence of the members of the crowd to conduct a literature review. Such competence can be demonstrated through "pre-flight" qualification tests that are administered as an entry criterion, before allowing workers to participate in the crowd. A second lesson is that it is important to decide on the scope of the literature review to ensure that the output of the literature review only includes sources relevant to the question.

Design for Crowdsourcing Literature Reviews

In this section, we describe a design for a leader-driven crowdsourcing platform that can be used to collect evidence and synthesize it into insights in a new domain. First, we identify the design principles that guide the design of the platform. Then, we present a conceptual model for crowdsourcing literature reviews. It includes both the structure of the artefacts produced by the crowd and the roles and responsibilities of the members of the crowd.

Design principles

The design of the proposed crowdsourcing platform builds on the lessons learned from the two manually conducted literature reviews in new domains and on recent advances in crowdsourcing. These lessons lead to seven design principles:

1. *Scoping and synthesis:* put a leader in charge to decide on which questions should be examined (scoping) and to synthesize the answers into new insights.
2. *Chunking:* partition the literature review task into focused microtasks that can be executed without having to consider the literature review as a whole.
3. *Diversity:* crowdsourcing benefits from having a diverse membership with different perspectives. Initially, it is assumed that crowd members cannot self-select to participate. The model, thereby, corresponds to the club of experts model of co-creation (Pater, 2009).
4. *Scaffolding:* embed expertise into the design of the tools to magnify worker efforts.
5. *Incremental points of reference:* show answers from other participants.
6. C*onsensus building:* create a consensus among the crowd members through commenting, voting, and tagging.
7. *Incentives:* build on the complementary motivations of leaders (to receive feedback) and contributors (to be recognized for their expertise).

Table 2 provides known uses in the crowdsourcing literature for each design principle.

Conceptual model

The design of the crowdsourcing platform proposed here draws on previous work on collaborative writing systems (Kim et al., 2014) and crowd-based clustering of documents (André et al., 2014). In a leader-driven crowdsourcing approach to collaborative writing (Kim et al., 2014), there are two types of participants: leaders and contributors. Leaders constrain and specify the nature of the contributions: the lead author of a literature review sets the scope of the literature review and guides the synthesis. Other crowd members (contributors) are recruited to focus on specific writing tasks.

As shown in Figure 1, we conceptualize a literature review as a story or narrative. Each narrative consists of a series of chunks that we call scenes (Kim et al., 2014). Each scene is anchored around a writing goal (such as providing an overview of the literature review, defining key features of the topic of the review, identifying examples illustrating the topic, or identifying gaps in the literature). Each writing goal is associated with a prompt that helps focus the work of the contributors. Answers to prompts are collected in drafts. Drafts can be commented and voted on, as well as categorized. It

Crowdsourcing Literature Reviews in New Domains

Michael Weiss

Table 2. Known uses of the design principles in the crowdsourcing literature

Design Principle	Known Use
Scoping and synthesis	In the Ensemble system, leaders decide what contributors should write by creating writing prompts and select drafts from the submitted drafts or write their own (Kim et al., 2014).
Chunking	In the Ensemble system, the writing tasks are distributed across different roles: leaders, moderators, and contributors (Kim et al., 2014). Lin and colleagues (2014) split satellite images into tiles and asked crowd members to tag individual tiles.
Diversity	In the Ensemble system, leaders reported that they found the perspectives of others beneficial for writing their narratives (Kim et al., 2014).
Scaffolding	Contributors write drafts for scenes in response to prompts created by leaders. When contributors write a draft, they can see all the drafts from other contributors for the same scene. This visibility provides context for their task (Kim et al., 2014). Scaffolding gives crowd members the right information, at the right location, and at the right time, to help them accomplish a given activity (Owens, 2013).
Incremental points of reference	In another study conducted by Kim and colleagues (2015), contributors are asked to create questions about social media posts. For each post, they are shown sample questions created by other contributors. In Lin et al. (2014), crowd members are shown all tags other crowd members have assigned to a geographical location. This visibility allows crowd members to compare their own decisions against those of other peer participants.
Consensus building	In André et al. (2014), crowd members iteratively categorize text fragments. When a crowd member is asked to categorize a fragment, they see how other fragments have been categorized. They can then decide to put the new fragment into an existing category or create a new one. In Kim et al. (2014), consensus is built by voting: leaders and contributors can vote on the drafts created by others.
Incentives	Leaders and collaborators have complementary motivations: leaders want to receive feedback on their narratives, whereas collaborators view their expertise as valuable input to leaders (Kim et al., 2014).

is up to the leader to choose the best draft for each scene, so as to produce a final version of the narrative.

Cognitive science research on writing has identified that the writing process can be viewed as a series of rhetorical problems (Flowers & Hayes, 1981). For each narrative, there is a top-level rhetorical problem (which includes the constraints given to the writers, and the goals the writers create for themselves), which is then decomposed into subproblems that drive the creation of the narrative. For example, if the lead author of a literature review needs input on examples illustrating the topic of the review, they can ask the contributors for specific contributions with a prompt. In this way, the lead author can maintain a high-level vision of the literature review, while providing contributors with enough context of the overall flow of the literature review and direction towards specific tasks to complete.

Table 3 lists the roles of the participants in a crowdsourced literature review process and their responsibilities. Note that, although terms such as scene, prompt, and draft are still generic, we expect to identify catalogs of scenes and prompts specific to the creation of literature reviews once an initial prototype of the proposed platform has been developed and can be subjected to

Crowdsourcing Literature Reviews in New Domains

Michael Weiss

Table 3. Roles and responsibilities in the crowdsourced literature review process.

Role	Narratives	Scenes	Drafts
Leader	Creates narratives	Creates scenes Creates prompts in scenes Determines sequence of scenes in a narrative	Creates drafts Synthesizes drafts into final draft for a narrative Comments on drafts Votes on drafts Categorizes drafts
Contributor			Creates drafts Comments on drafts Votes on drafts Categorizes drafts

systematic user testing. For example, it is already apparent from the experience with the two manually created literature reviews that the platform will need to support different types of drafts.

In one case, a prompt may ask contributors to produce alternatives to the leader's draft. For example, the leader could ask contributors for a definition of software lineage for malware. In this case the drafts are strictly alternative versions of a scene. In another case, a prompt could ask for a list of instances that together form the answer to the question. For example, a leader might ask for examples of code reuse attacks and for contributors to categorize them (André et al., 2014). As contributors collect and categorize the examples, they produce a taxonomy of code reuse attacks that could serve as a basis for further exploration. In this case, all or a subset of the drafts should be included in the review.

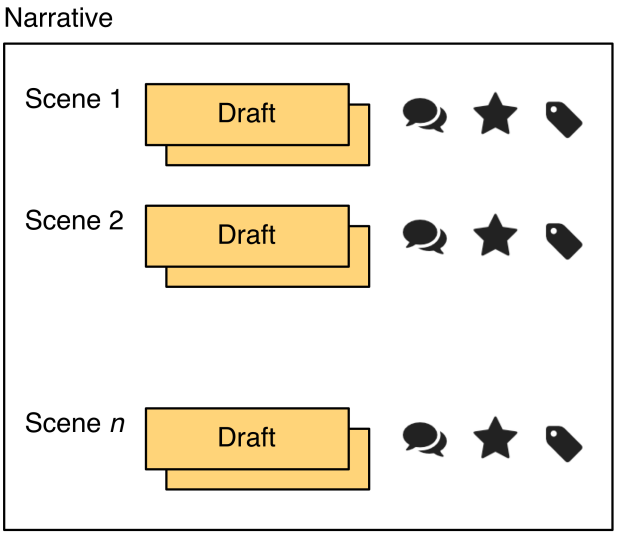

Figure 1. Conceptualization of a literature review as a narrative, the components of a literature review (scenes and drafts), and the actions that can be performed on each of the components (commenting, voting, and categorization)

Conclusion

In this article, we proposed the design of a platform for crowdsourcing literature reviews in new domains. In particular, our focus was on creating systematic narrative literature reviews. Benefits expected from crowdsourcing literature reviews include:

1. Reducing the time it takes to complete a review

2. Being able to rely on emergent validation criteria given that a new domain lacks a pre-existing reference

Crowdsourcing Literature Reviews in New Domains

Michael Weiss

for what constitute anomalies that may indicate gaps in current knowledge

3. Leveraging the diversity of perspectives of crowd members

4. Limiting the level of specific domain knowledge required to create a literature review in a new domain

Challenges for crowdsourcing literature reviews that we foresee include:

1. How to encourage participation (what kind of incentives need to be provided)

2. How to ensure the quality of the reviews produced (what aspects of the crowdsourcing process should be instrumented)

3. How to further support the synthesis stage of the review (what role can advanced techniques such as visualization and text mining techniques)

A prototype of the platform is currently being implemented by a team of developers at VENUS Cybersecurity Corporation (venuscyber.com). Systematic user testing of the platform and resulting extensions to the platform are left for future work.

Acknowledgements

This work was conducted for VENUS Cybersecurity Corporation and has received support from the Canadian Security Establishment (CSE) and the Laboratory of Analytical Sciences (LAS) at North Carolina State University. I would like to thank the authors of the two literature reviews who provided the empirical basis for the proposed process to crowdsourcing literature reviews for new domains (in alphabetical order: Tony Bailetti, Dan Craigen, Mahmoud Gad, and Ahmed Shah). The proposed design also benefited greatly from discussions with the team at Venus tasked to implement the crowdsourcing platform (in alphabetical order: Ibrahim Abualhaol, Ali Abu Alhawa, Mohamed Amin, Chris Budiman, and Raed Iskander).

About the Authors

Michael Weiss holds a faculty appointment in the Department of Systems and Computer Engineering at Carleton University in Ottawa, Canada, and is a member of the Technology Innovation Management program. His research interests include open source, ecosystems, mashups, patterns, and social network analysis. Michael has published on the evolution of open source business, mashups, platforms, and technology entrepreneurship.

References

André, P., Kittur, A., & Dow, S. P. 2014. Crowd Synthesis: Extracting Categories and Clusters from Complex Data. *Proceedings of the ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW):* 989–998. New York: ACM. http://dx.doi.org/10.1145/2531602.2531653

Attenberg, J., Ipeirotis, P. G., & Provost, F. J. 2011. Beat the Machine: Challenging Workers to Find the Unknown Unknowns. *Human Computation: Papers from the 2011 AAAI Workshop (WS-11-11):* 2–7.

Baumeister, R. F., & Leary, M. R. 1997. Writing Narrative Literature Reviews. *Review of General Psychology,* 1(3): 311–320. http://dx.doi.org/10.1037/1089-2680.1.3.311

Berry, D. 1995. The Importance of Ignorance in Requirements Engineering. *Journal of Systems and Software,* 28(1): 179–184. http://dx.doi.org/10.1016/0164-1212(94)00054-Q

Bigham, J. P., Bernstein, M. S., & Adar, E. 2015. Human-Computer Interaction and Collective Intelligence. In T. Malone & M. Bernstein (Eds.). H*andbook of Collective Intelligence:* 57–83, MIT Press. http://cci.mit.edu/CIchapterlinks.html

Brown, A. W., & Allison, D. B. 2014. Using Crowdsourcing to Evaluate Published Scientific Literature: Methods and Example. *PloS ONE,* 9(7): e100647. http://dx.doi.org/10.1371/journal.pone.0100647

Elliot, J., Thomas, J., & Owens, N., Dooley, G., Riis, J., Wallace, B., Thomas, J., Noel-Storr, A., Rada, G., Struthers, C., Howe, T., MacLehose, H., Brandt, L., Kunnamo, I., & Mavergames, C. 2014. Editorial: #CochraneTech: Technology and the Future of Systematic Reviews. *Cochrane Library,* 2014(9). http://dx.doi.org/10.1002/14651858.ED000091

Embase. 2016. Embase Screening. *The Cochrane Collaboration.* Accessed on February 7, 2016: http://screening.metaxis.com/EMBASE/login.php

Flower, L., & Hayes, J. R. 1981. A Cognitive Process Theory of Writing. *College Composition and Communication,* 32(4): 365–387. http://www.jstor.org/stable/356600

Grant, M. J., & Booth, A. 2009. A Typology of Reviews: An Analysis of 14 Review Types and Associated Methodologies. *Health Information & Libraries Journal,* 26(2): 91–108. http://dx.doi.org/10.1111/j.1471-1842.2009.00848.x

Crowdsourcing Literature Reviews in New Domains

Michael Weiss

Green, B. N., Johnson, C. D., & Adams, A. 2006. Writing Narrative Literature Reviews for Peer-Reviewed Journals: Secrets of the Trade. *Journal of Chiropractic Medicine,* 5(3): 101–117. http://dx.doi.org/10.1016%2FS0899-3467(07)60142-6

iHub Research. 2013. *Viability, Verification, Validity: 3Vs of Crowdsourcing.* Nairobi, Kenya: iHub Research. http://www.ihub.co.ke/ihubresearch/jb_VsReportpdf2013-8-29-07-38-56.pdf

Kim, J., Cheng, J., & Bernstein, M. S. 2014. Ensemble: Exploring Complementary Strengths of Leaders and Crowds in Creative Collaboration. *Proceedings of the ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW):* 745–755. New York: ACM. http://dx.doi.org/10.1145/2531602.2531638

Lin, A. Y. M., Huynh, A., Lanckriet, G., & Barrington, L. 2014. Crowdsourcing the Unknown: The Satellite Search for Genghis Khan. *PloS ONE,* 9(12): e114046. http://dx.doi.org/10.1371/journal.pone.0114046

Luther, K., Hahn, N., Dow, S. P., & Kittur, A. 2015. Crowdlines: Supporting Synthesis of Diverse Information Sources through Crowdsourced Outlines. *Third AAAI Conference on Human Computation and Crowdsourcing,* November 8-11, San Diego, CA. Palo Alto, CA: The AAAI Press.

Mehrotra, G. and Berry, D.M. 2012. *Role of Domain Ignorance in Software Development: Software Development Tasks Benefiting from Domain Ignorance.* Technical Report, Cheriton School of Computer Science, University of Waterloo, Canada. https://cs.uwaterloo.ca/~dberry/FTP_SITE/tech.reports/MehrotraBerrySurvey.pdf

Owens, T. 2013. Digital Cultural Heritage and the Crowd. Curator: *The Museum Journal,* 56(1): 121–130. http://dx.doi.org/10.1111/cura.12012

Paré, G., Trudel, M. C., Jaana, M., & Kitsiou, S. 2015. Synthesizing Information Systems Knowledge: A Typology of Literature Reviews. I*nformation & Management,* 52(2): 183–199. http://dx.doi.org/10.1016/j.im.2014.08.008

Pater, M. 2009. Co-Creation's 5 Guiding Principles. *Fronteer Strategy:* April 9, 2009. Accessed February 1, 2016: http://fronteerstrategy.blogspot.ca/2009/04/co-creations-5-guiding-principles-or.html

Citation: Weiss, M. 2016. Crowdsourcing Literature Reviews in New Domains. *Technology Innovation Management Review*, 6(2): 5–14. http://timreview.ca/article/963

Keywords: literature review, new domains, systematic, narrative, crowdsourcing, crowdsourcing platform, co-creation, cybersecurity

Intrusion Learning: An Overview of an Emergent Discipline

Tony Bailetti, Mahmoud Gad, and Ahmed Shah

> *"The illiterate of the 21st Century are not those who cannot read and write but those who cannot learn, unlearn and relearn.*
>
> Alvin Toffler
> Writer and futurist
> In *Powershift*

The purpose of this article is to provide a definition of intrusion learning, identify its distinctive aspects, and provide recommendations for advancing intrusion learning as a practice domain. The authors define intrusion learning as the collection of online network algorithms that learn from and monitor streaming network data resulting in effective intrusion-detection methods for enabling the security and resiliency of enterprise systems. The network algorithms build on advances in cyber-defensive and cyber-offensive capabilities. Intrusion learning is an emerging domain that draws from machine learning, intrusion detection, and streaming network data. Intrusion learning offers to significantly enhance enterprise security and resiliency through augmented perimeter defense and may mitigate increasing threats facing enterprise perimeter protection. The article will be of interest to researchers, sponsors, and entrepreneurs interested in enhancing enterprise security and resiliency.

Introduction

Intrusion learning offers the potential of significantly improving the security and resiliency of enterprise systems and increase the enterprise's capability to adapt to adversaries and changes in business environments. This article positions the emerging domain of intrusion learning at the intersection of machine learning, intrusion detection, and streaming network data. Machine learning refers to the algorithms that are first trained with reference input to "learn" its specifics, to then be deployed on previously unseen input for the actual detection process (Sommer & Paxson, 2010). Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices (Scarfone & Mell, 2007). By streaming network data, we mean streams of distinct and diverse network events flowing on a network over time. This definition is consistent with the definition of data stream provided by Savvius (2016).

We draw upon the results of a literature review carried out for the purpose of defining intrusion learning. We start with a summary of the literature review and then define intrusion learning, identify its distinctive aspects, and provide recommendations for advancing the emerging discipline. We end with our conclusions.

Literature Review

We performed a systematic narrative review to identify the latest advancements published in the academic literature with respect to machine learning, streaming network data, and intrusion detection. Articles in English-language journals published from 2010 to 2015 in North America and Europe were reviewed. We organized the literature into five themes: i) feature extraction, ii) learning algorithms, iii) clustering, iv) datasets, and v) tools.

Intrusion Learning: An Overview of an Emergent Discipline

Tony Bailetti, Mahmoud Gad, and Ahmed Shah

Feature extraction
Feature extraction is the process of determining a subset of features from an original set. The intent of feature extraction is to find a combination of original features or data attributes that can better describe the internal structure of the data. The three principal algorithms that are used for feature extraction are: locality preserving projection (linear projective maps arising from solving a variational problem optimally preserving neighbourhood structure), linear discriminate analysis (a method for finding a linear combination of variables that optimally separates classes) and principle component analysis (a linear technique that projects the data along the directions of maximal variance) (Fisher, 1936; He, 2005; Parakash & Surendran, 2013).

Intrusion detection systems use feature extraction to determine what features or attributes can assist with detecting malicious traffic (Laxhammer, 2014). We found two feature extraction challenges in the context of streaming network data. First, the dynamic changing nature of the streams results in challenges pertaining to the evolution of features (the emergence of new features), concept evolution (new classes evolving into the stream), and concept drift (underlying concepts change) (Momin & Hambir, 2015). The second challenge is that data streams are, in principle, of infinite length (Masud et al., 2010). Most existing data stream classification techniques address only the infinite length and concept-drift problems; concept evolution and feature evolution are ignored. In the face of a dynamic adversary, ignoring concept evolution and feature evolution increases enterprise risk.

Learning algorithms
Three emerging machine-learning algorithms play important roles in intrusion learning: active learning, adversarial learning, and conformal prediction. Active learning is a subfield of artificial intelligence and machine learning, and it refers to the study of computer systems that improve with experience and training (Settles, 2012). Adversarial learning refers to the study of effective machine learning techniques against an adversarial opponent (Huang et al., 2011). Conformal prediction refers to hedging individual predictions made by machine learning algorithms with valid measures of confidence (Laxhammar & Falkman, 2011).

The presence of an adversary changes the dynamics for learning algorithms. An adversary will attempt to poison or manipulate the data so that the algorithms treat the malicious as benign. This adversarial context has led to research on how algorithms can unlearn poisoned and polluted data (Cao & Yang, 2015).

Clustering
Organizing data into sensible groupings is one of the most fundamental modes of understanding and learning (Jain, 2010). Clustering is used to detect unknown attacks and discover unusual activities or usage patterns in traffic data in real time. The value of clustering comes from discovering groups and structures in the data that, in some way, are similar to each other, without prior knowledge of the data structures.

Data stream algorithms can only read the incoming data once and must do so in the context of having to respond in real-time with bounded memory usage. These algorithms can only provide approximate results and must support evolving concepts (Nguyen & Luo, 2013).

Because real-time data streams are unbounded, it will only be possible to process a portion of the entire data stream one "window" at a time (Nguyen & Luo, 2013). Various kinds of windows-based algorithms exist. For example, the sliding window algorithm analyzes the most recent data points and is suitable for applications where only the most recent information is of interest. The main disadvantage is that it ignores parts of the data streams. An adversary could manipulate a sliding window so that malicious activities occur in those parts of the streams being ignored by the algorithm.

Datasets
A dataset contains network traffic that is used to benchmark the performance of network intrusion algorithms. Datasets may include a combination of malicious traffic, non-malicious traffic, and identified features that can be used for testing. The most commonly used dataset researchers use for intrusion detection dates back to the KDD Cup 1999 (archive.ics.uci.edu/ml/datasets.html). It is surprising that a dataset from 1999 is still commonly used given the significant changes in attack tools, techniques, and data types that have occurred since then.

That the KDD Cup 1999 dataset is still used suggests that developing or accessing contemporary datasets is a major challenge. Privacy rights, confidentiality, and intellectual property are all concerns that impede access to real network data. Though there are other datasets available, the reality is that valid contemporary streaming data is unavailable outside of large Internet providers. The absence of new datasets retards science-based experimentation of new algorithms.

Intrusion Learning: An Overview of an Emergent Discipline

Tony Bailetti, Mahmoud Gad, and Ahmed Shah

Tools

Many publicly available experiments that are applying machine learning to intrusion detection are using a tool called massive online analysis (MOA; moa.cms.waikato.ac.nz). MOA is a machine-learning framework that contains real-time stream processing algorithms. It is not customizable for multi-node and scalable distributable processing.

However, scalable and distributable machine-learning processing engines that can process real-time streaming information do exist (e.g., SAMOA; samoa.incubator.apache.org). However, they have not been widely found in streaming intrusion-detection machine-learning experiments. We have not determined why this situation exists, though we note that SAMOA is a relatively new Apache project. SAMOA is one of few open source tools that is specifically designed for distributed and true real-time streaming (Landset et al., 2015). Apache Spark with MLib also includes a distributed architecture for processing data streams (spark.apache.org).

Defining Intrusion Learning

In this section, we propose a definition of intrusion learning based upon four elements: i) the ultimate outcome of intrusion learning; ii) the target of the ultimate outcome; iii) the mechanism used to deliver the ultimate outcome; and iv) the interdependence between intrusion learning and scientific and technological advances.

We propose the following definition of intrusion learning:

> *Intrusion learning is the collection of online network algorithms that learn from and monitor streaming network data resulting in effective intrusion detection methods for enabling the security and resiliency of enterprise systems. The network algorithms build on advances in cyber-defensive and cyber-offensive capabilities.*

We characterized the elements underpinning this definition as follows:

1. *Ultimate outcome:* Effective intrusion-detection methods on streaming network data.

2. *Target of ultimate outcome:* Security and resiliency of enterprise systems is the key target outcome.

3. *Mechanism used to deliver ultimate outcome:* Online network algorithms that learn from and monitor streaming network data.

4. *Interdependence of this mechanism from scientific and technological advances*: The mechanisms must build upon advances in both cyber-defensive and cyber-offensive capabilities (e.g., new machine-learning algorithms, new attack vectors), which themselves are informed by multi-disciplinary thinking.

Distinctive Aspects

We believe that there are five distinctive aspects of the intrusion learning domain relative to the machine learning, intrusion detection, and streaming domains:

1. *Real-time analysis of streaming network data:* Intrusion learning must respond to intrusions in real time. Unlike big data analytics, intrusion learning requires approximations, windowing, and other techniques to produce effective timely scalable analysis of network data (Aggarwal, 2007).

2. *High cost of failure:* The cost of failure of machine-learning algorithms is much higher for intrusion detection (e.g., loss of intellectual property and brand damage) compared to other applications of machine learning such as optical character recognition (Sommer & Paxson, 2010).

3. *Adversarial context:* Intrusion learning must deal with the existence of talented and determined adversaries. The presence of the adversary requires that intrusion learning must evolve with ongoing advances in both cyber-defensive and cyber-offensive capabilities (Cao & Yang, 2015; Corona et al., 2013).

4. *Network traffic diversity:* Intrusion learning must deal with the variability of network traffic (e.g., bandwidth, load balancing, and connection requests). Traffic diversity complicates the perspective of "normal" and therefore hinders the ability to identify an anomaly (Sommer & Paxson, 2010).

5. *Outlier detection:* Machine-learning algorithms are better at finding similarities than anomalies. As noted by Sommer and Paxon (2010), "the classic machine learning application is a classification problem, rather than discovering meaningful outliers as required by an anomaly detection system."

Intrusion Learning: An Overview of an Emergent Discipline

Tony Bailetti, Mahmoud Gad, and Ahmed Shah

Recommendations

The recommendations that follow are directed at researchers, sponsors and entrepreneurs interested in intrusion learning:

1. *Understand the threat model.* For example, researchers must know the cost of missed attacks (Sommer & Paxson, 2010).

2. *Learn, unlearn, and relearn.* Adversaries will act to mislead algorithms by steering the analyses to recognize the malicious as benign. Effective responses to such attacks need development. Corona and colleagues (2013) examine adversarial attacks against intrusion-detection systems as well as related taxonomies and potential solutions to known issues. This perspective leads to the concept of systems "unlearning" or forgetting what they had incorrectly "learned" (Cao & Yang, 2015).

3. *Select a narrow research scope.* The objectives of the research must be concrete. For example, researchers should determine precisely what kinds of attacks are being detected and what techniques are to be applied. The research should be able to answer such questions as to what attacks are being detected and the reasons as to why the attacks are being recognized (Sommer & Paxson, 2010).

4. *Develop new datasets.* To advance intrusion learning as a domain of practice, new datasets reflecting current network traffic need to be developed. For evidence-based evaluations, it is crucial to experiment with real datasets while observing societal norms such as privacy and commercial concerns.

5. *Develop open source intrusion learning tools that can scale.* Researchers need access to scalable machine learning tools. Although scalable proprietary tools exist, researchers worldwide must have access to tools that are capable of analyzing the reality of today's network traffic. Intrusion learning cannot advance in the absence of scalable machine learning tools.

6. *Improve online analytics.* Intrusion learning requires a combination of online and offline analyses. To properly enable real-time intrusion responsiveness, the balance between online and offline analytics needs to lean more heavily towards the online.

7. *Automate responses.* It is all very well to recognize the presence of anomalous or malicious activities. However, there is a need to go one step further and embed intrusion learning into the enterprise controllers. With highly scalable and changeable attacks, defensive responses must react in kind.

8. *Anticipate attacks.* By observing adversary community dynamics, it may be possible to anticipate attacks and react accordingly. Such research would move the discovery and detection outside the enterprise perimeter.

9. *Enhance feature extraction.* Research should aim to expand the set of extractable features that correlate with malicious traffic. This research could remain at the level of network flow, but richer theories are likely to provide more substantial payoffs.

Conclusion

In this article, we introduced the concept of intrusion learning as a domain that draws from machine learning, intrusion detection, and streaming network data. A key benefit of intrusion learning is that it may significantly enhance enterprise security and resiliency through augmented perimeter defense.

We identified a set of unique attributes and recommendations for advancing intrusion learning. For intrusion learning to meet its objectives of enhanced security and resiliency, these recommendations should not be treated in isolation but build upon each other: cross-cutting thinking (over machine learning, intrusion detection, and streaming) that focuses upon the distinctive aspects of intrusion learning will enhance progress.

Perhaps our most important recommendation is the development of new datasets that reflect contemporary network data and malware. The absence of such datasets is a significant impediment to the validation of intrusion-learning techniques. Privacy rights, confidentiality, etc., are concerns that are impeding the development of such datasets. We end this article with a "call to action" to develop such datasets, properly informed by researchers, privacy advocates, policy personnel, and so on, so that societal concerns are addressed.

Intrusion Learning: An Overview of an Emergent Discipline

Tony Bailetti, Mahmoud Gad, and Ahmed Shah

Acknowledgements

The authors thank Dan Craigen, Science Advisor at the Communications Security Establishment and a Visiting Scholar in the Technology Innovation Management program, for his invaluable input into the development and refinement of this article.

About the Authors

Tony Bailetti is an Associate Professor in the Sprott School of Business and the Department of Systems and Computer Engineering at Carleton University, Ottawa, Canada. Professor Bailetti is the Director of Carleton University's Technology Innovation Management (TIM) program. His research, teaching, and community contributions support technology entrepreneurship, regional economic development, and international co-innovation.

Mahmoud M. Gad is a Research Associate at VENUS Cybersecurity. He holds a PhD in Electrical and Computer Engineering from the University of Ottawa in Canada. Additionally, he holds an MSc in Electrical and Computer Engineering from the University of Maryland in College Park, United States. His research interests include cybercrime markets, machine learning for intrusion detection, analysis of large-scale networks, and cognitive radio networks.

Ahmed Shah holds a BEng in Software Engineering and is pursuing an MASc degree in Technology Innovation Management at Carleton University in Ottawa, Canada. Ahmed has experience working in cybersecurity research with the VENUS Cybersecurity Corporation and has experience managing legal deliverables at IBM.

References

Aggarwal, C. (Ed.) 2007. Data Streams: Models and Algorithms. New York: Springer.

Cao, Y., & Yang, J. 2015. Towards Making Systems Forget with Machine Unlearning. In *Proceedings of the 2015 IEEE Symposium on Security and Privacy:* 463–480. New York, IEEE. http://dx.doi.org/10.1109/SP.2015.35

Corona, I., Giacinto, G., & Roli, F. 2013. Adversarial Attacks Against Intrusion Detections Systems: Taxonomy, Solutions and Open Issues. *Information Science,* 239: 201–225. http://dx.doi.org/10.1016/j.ins.2013.03.022

Fisher, R. A. 1936. The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics, 7(2): 179–188. http://dx.doi.org/10.1111/j.1469-1809.1936.tb02137.x

He, X. 2005. *Locality Preserving Projections.* Doctoral thesis, University of Chicago.

Huang, L., Joseph, A. D., Nelson, B., Rubinstein, B. I. P., & Tygar, J. D. 2011. Adversarial Machine Learning. In *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence:* 43–58. http://dx.doi.org/10.1145/2046684.2046692

Jain, A. 2010. Data Clustering: 50 Years Beyond K-means. *Pattern Recognition Letters,* 31(8): 651–666. http://dx.doi.org/10.1016/j.patrec.2009.09.011

Landset, S., Khoshgoftaar, T. M., Richter, A. N., & Hasanin, T. 2015. A Survey of Open Source Tools for Machine Learning with Big Data in the Hadoop Ecosystem. *Journal of Big Data,* 2(1): 1–36. http://dx.doi.org/10.1186/s40537-015-0032-1

Laxhammer, R. 2014. *Conformal Anomaly Detection: Detecting Abnormal Trajectories in Surveillance Applications.* Doctoral Thesis, University of Skövde School of Informatics, Sweden.

Laxhammar, R., & Falkman, G. 2011. Sequential Conformal Anomaly Detection in Trajectories Based on Hausdorff Distance. In *Proceedings of the 14th International Conference on Information Fusion.* New York, IEEE.

Masud, M., Chen, Q., Guo, J., Khan, L., & Han, J., Thuraisingham, B. M. 2010. Classification and Novel Class Detection of Data Streams in a Dynamic Feature Space. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases:* 337–352, 2010. http://dx.doi.org/10.1109/TKDE.2010.61

Momin, N., & Hambir, N. 2015. A Survey on Various Classification and Novel Class Detection Approaches for Feature Evolving Data Stream. *Multidisciplinary Journal of Research in Engineering and Technology,* 2(1): 342–346.

Nguyen, K., & Luo, Z. 2013. Reliable Indoor Location Prediction Using Conformal Prediction. *Annals of Mathematics and Artificial Intelligence,* 74(1): 133–153. http://dx.doi.org/10.1007/s10472-013-9384-4

Parakash, D., & Surendran, S. 2013. Detection and Analysis of Hidden Activities in Social Networks. *International Journal of Computer Applications,* 77(16): 34–38. http://dx.doi.org/10.5120/13570-1404

Savvius. 2016. Glossary of Networking Terms. *Savvius.* Accessed February 15, 2016: http://www.wildpackets.com/resources/compendium/glossary_of_networking_terms#S

Intrusion Learning: An Overview of an Emergent Discipline

Tony Bailetti, Mahmoud Gad, and Ahmed Shah

Scarfone, K., & Mell, P. 2007. *Guide to Intrusion Detection and Prevention Systems (IDPS)*. NIST Special Publication 800-94. Gaithersburg, MD: National Institute of Standards and Technology.

Settles, B. 2012. Active Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning,* 6(1): 1–114. http://dx.doi.org/10.2200/S00429ED1V01Y201207AIM018

Sommer, R., & Paxson, V. 2010. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy:* 305–316. http://dx.doi.org/10.1109/SP.2010.25

Citation: Bailetti, T., Gad, M., & Shah, A. 2016. Intrusion Learning: An Overview of an Emergent Discipline. *Technology Innovation Management Review*, 6(2): 15–20. http://timreview.ca/article/964

Keywords: cybersecurity, intrusion learning, intrusion detection, machine learning, learning algorithms, adversarial learning, clustering, streaming network data, real-time analysis, enterprise, security, resiliency

Examining the Modes Malware Suppliers Use to Provide Goods and Services

Tony Bailetti and Mahmoud Gad

> *"Remove the predators, and the whole ecosystem begins to crash like a house of cards."*
>
> Brian Skerry
> Underwater photojournalist

Malware suppliers use various modes to provide goods and services to customers. By mode, we mean "the way" the malware supplier chooses to function. These modes increase monetization opportunities and enable many security breaches worldwide. A theoretically sound framework that can be used to examine the various modes that malware suppliers use to produce and sell malware is needed. We apply a general model specified recently by Hagiu and Wright to study five modes that malware suppliers use to deliver goods and services to their customers. The framework presented in this article can be used to predict the mode in which a malware supplier will function; to study which types of malware suppliers, agents, and customers are attracted to each mode; to discover new modes; and to better understand the threat a malware supplier presents.

Introduction

Malware suppliers, agents, and customers play important roles in the cybercrime economy. Malware suppliers include technically skilled individuals who produce and distribute malicious code; agents who act on behalf of malware suppliers or directly interact with customers; and customers who purchase goods and services to gain unauthorized access to compromised computers' data and resources, steal e-currency, exfiltrate victims' personal information, and so on (Kamluk, 2009).

The modes that malware suppliers use to provide goods and services to customers increase illicit monetization opportunities and enable many of the recent security breaches that have targeted some of the largest financial, government, military, and retail institutions in the world (Ablon et al., 2014; Armin, 2013; Gu, 2013; Samani, 2013). However, it is difficult to understand what these modes have in common, what makes them different, and what their potential combinations may be.

Consider the following examples of malware supplier modes:

1. *Dark0de:* a multisided platform that served as a venue for the sale and trade of hacking services, botnets, malware, and other illicit goods and services from 2007 until July 2015 when it was shut down by the Federal Bureau of Investigations (Europol, 2015). It took only two weeks for this marketplace to start operating again (Clark, 2015; Kovacs, 2015).

2. *Power Locker:* a reseller that allows customers to customize ransomware (Goodin, 2014; Mathews, 2014).

3. *Hacking Team* (hackingteam.it): a Milan-based firm that focuses on all aspects of offensive cybersecurity. On July 8, 2015, WikiLeaks released more than one million searchable emails from this Italian surveillance malware vendor (WikiLeaks, 2015). Moreover, the source code for Hacking Team's flagship software, Remote Control System, was breached and used to attack websites in South Korea (Peters, 2015; The Chosunilbo, 2015).

4. *The Styx Exploit Pack:* a kit vendor that sells a high-end software package developed for "the underground" but is marketed and serviced online. A 24-hour virtual help desk is available to paying customers (Krebs, 2013).

Examining the Modes Malware Suppliers Use to Provide Goods and Services

Tony Bailetti and Mahmoud Gad

These examples illustrate that the lack of a theoretically-grounded framework to examine the nuances of the modes in which malware suppliers function hinders understanding of how the cybercrime economy works and weakens mitigation strategies.

The choice a firm makes about the mode it uses to deliver goods and services to customers is relevant in many product markets because of the increase in the number and size of online marketplaces that have emerged recently (Edelman, 2015; Hagiu, 2007; Hagiu & Wright, 2013, 2015b). Moreover, the choice of mode a malware supplier uses to deliver goods and services is prominent in a market where advances in obfuscation and detection-avoidance techniques, software reuse, machine learning, and Internet and mobile technologies have made it possible to use various approaches that offer an increasing variety of malware goods and services to customers.

The literature on the different modes in which a firm can function can be organized based on the methods used to examine them: specification of formal general models (Boudreau & Hagiu, 2009; Hagiu, 2007; Hagiu, 2009; Hagiu & Wright, 2015a; 2015b, 2015c); empirical studies (Boudreau, 2010); and informal descriptions (Choudary, 2015; Edelman, 2015; Eisenmann et al., 2006; Hagiu, 2014; Hagiu & Wright, 2013). This study focuses on five modes in which a firm can operate that have been specified using formal general models: "employment", "multisided platform", "reseller", "vertically integrated", and "input-supplier" (Hagiu, 2007; Hagiu, 2009; Hagiu & Wright, 2015a, 2015b, 2015c).

In the remainder of this article, we summarize the general model developed by Hagiu and Wright (2015c) to examine the choice a firm with a single agent makes among alternate modes to deliver goods and services and then apply the general model to examine five approaches that we believe malware suppliers use to provide products and services to their customers. We then discuss the contribution of this research and provide conclusions.

General Model with One Firm and One Agent

The general model for a firm and a single agent developed by Hagiu and Wright (2015c) assumes that the revenue generated jointly by the firm and the agent depends on three types of actions, all of which are influenced by asset ownership. These actions are referred to as being non-contractible. The non-contractible actions can be organized into three types: i) actions that can solely be carried out by the firm, ii) actions that can solely be carried out by the agent, and iii) transferable actions that can be carried out by either the firm or the agent.

The firm and the agent incur costs carrying out their actions. These costly actions are expected to increase the revenue generated jointly by the firm and the agent. Any contract offered by the firm to the agent can only depend on the revenue generated by the three types of actions, not just one or two types. The firm can offer the agent a contract that consists of a fixed fee and a variable fee equal to a percentage of the revenue generated jointly by the firm and the agent. The firm or the agent can collect revenues and pay the other party their share.

Hagiu and Wright (2015c) examine the case where a firm can select to operate in one of two modes: "employment" and "multisided platform". The difference between the two modes is that the firm controls the transferable actions in the "employment" mode and the agent controls the transferable actions in the "multisided platform" mode. A side refers to an actor type. For example, a two-sided platform may enable individuals seeking employment and employers to interact directly. Similarly, a multisided platform may enable service providers, customers, and customers' customers to interact directly.

According to Hagiu and Wright (2015b), two features make the multisided platform mode special. First, the multisided platform enables direct interactions between agents and customers. The phrase "direct interactions" is used to mean that the agent and the customers, not the firm, retain control over the key terms of the interaction. These terms can include price, bundling, delivery, quality, and so on.

The second feature that makes the multisided platform special is that both the agent and the customers are affiliated to the multisided platform. Agents make cash and in-kind investments in the multisided platform to interact with customers and form expectations of future returns from these investments. Similarly, anticipating returns, customers make cash and in-kind investments in the multisided platform to interact directly with the agent.

Examining the Modes Malware Suppliers Use to Provide Goods and Services

Tony Bailetti and Mahmoud Gad

Examining Modes Used by a Malware Supplier with One Agent

Consider the case where there is one malware supplier, one agent, one or more customers, and one or more customers' customers. Assume that the malware supplier is a technical organization with malware goods and services as its output. To produce and sell malware to customers, the malware supplier needs to choose one of the five modes illustrated in Figure 1:

1. *Employment mode:* employ and incentivize an agent to provide goods and service to customers
2. *Multisided platform mode:* enable the affiliated agent to provide goods and services directly to affiliated customers
3. *Reseller mode:* buy from a seller and resell to customers
4. *Vertically integrated mode:* work for a vertically integrated organization
5. *Input supplier mode:* sell inputs to a kit vendor who in turn incorporates those inputs in goods and services they sell to their customers

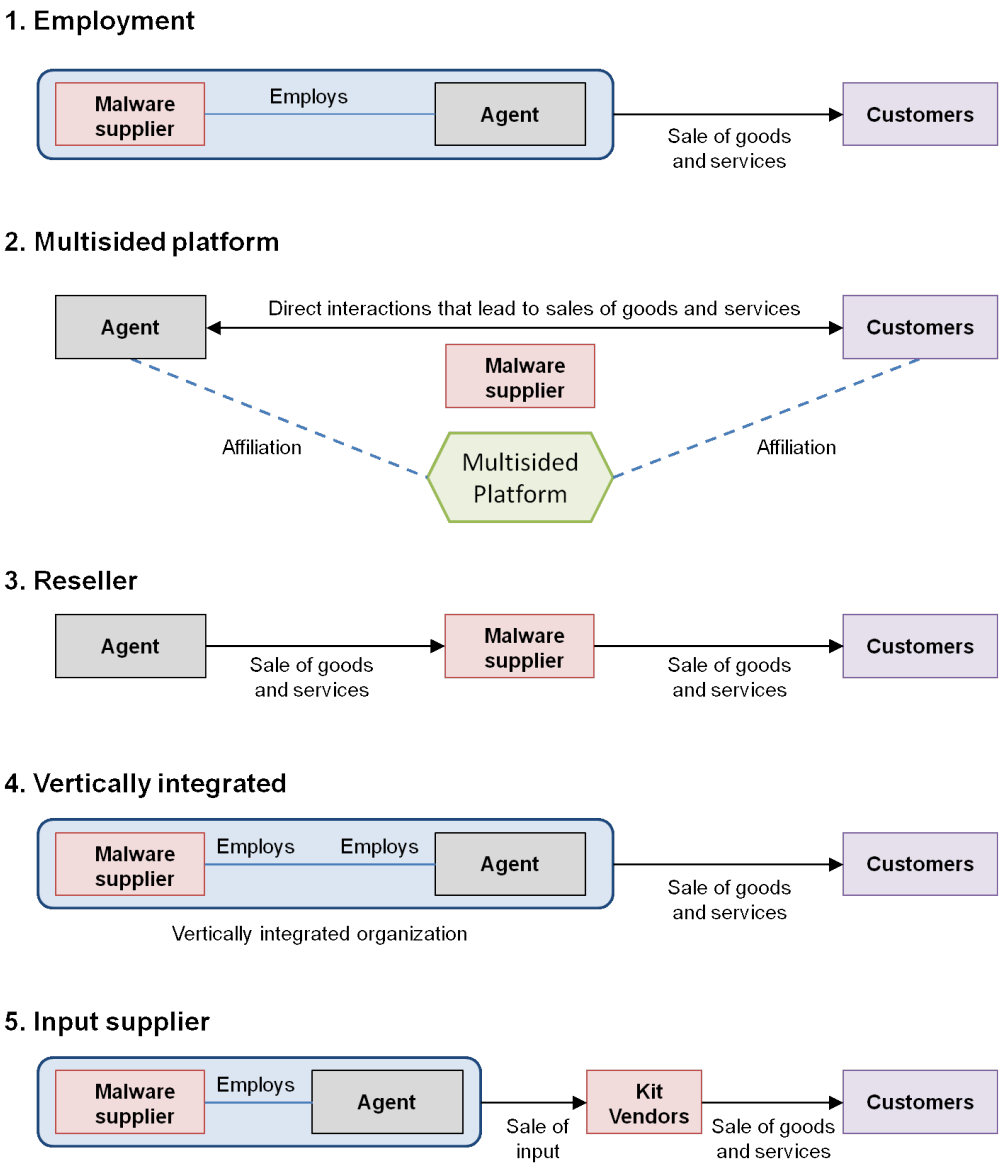

Figure 1. Modes to supply malware goods and services to customers

Examining the Modes Malware Suppliers Use to Provide Goods and Services

Tony Bailetti and Mahmoud Gad

Further assume that the role of the agent is the same in all five modes: help monetize the output of the malware supplier. In the "employment" mode, the agent is an employee of the malware supplier. In the "multisided platform" mode, the agent is an affiliated independent professional who is enabled by the malware supplier's multisided platform to provide goods and services directly to customers. In the "reseller" mode, the agent (shown as Seller in Figure 1) sells those goods and services to the malware supplier that their customers wish to purchase. In the "vertically integrated" mode, both the agent and the malware supplier are employees of the same organization. In the "input supplier" mode, the agent is either an employee of the malware supplier or has no role. The malware supplier sells inputs to kit vendors, and these inputs become part of the goods and services kit vendors sell to customers located downstream in the value chain.

Table 1 provides an example of non-contractible actions organized into the three action types identified in the previous section. Note that the information on Table 1 depends on the role of the agent. Recall that, in our example, the agent's role is to help monetize the outputs of the malware supplier. If the role of the agent was a technical one, the information in rows denoted 2 and 3 in Table 1 would be different.

The non-contractible actions that can solely be carried out by the malware supplier are those which are part of an ongoing investment in the firm. These actions are non-transferable. The non-contractible actions that can solely be carried out by the agent are those that are part of an ongoing effort made by the agent in the provision of its service. These actions are also non-transferable.

Table 1. Non-contractible actions by type

Action Type	Possible Non-Contractible Actions
1. Actions that can be carried out solely by the malware supplier	• Design and maintain the system to avoid detection • Maintain and upgrade code and techniques to reuse in attack approach • Operate specialized equipment to write and test new code as well as integrate new and reused code • Control code versions and modules • Design information and communications infrastructure • Automate the exploitation of client-side vulnerabilities (e.g., target browsers and programs that a website can invoke through the browser)
2. Actions that can be carried out solely by the agent	• Market and sell • Manage service quality • Develop new distribution and sales channels
3. Transferable actions that can be carried out by the malware supplier or the agent	• Assemble and update information about product–market fit • Support customers • Target vulnerabilities to exploit • Train customers and intermediaries • Promote in the underworld • Avoid detection of off-line operations • Arrange for escrow payments • Leverage others' communications infrastructures (e.g., botnets) and people networks

Examining the Modes Malware Suppliers Use to Provide Goods and Services

Tony Bailetti and Mahmoud Gad

Non-contractible actions that can be carried out by the malware supplier or the agent are referred to as transferable actions. In the general model with one agent developed by Hagiu and Wright (2015c), the mode in which the malware supplier operates depends on whether the firm or the agent controls the transferable actions. In our example, if the malware supplier chooses to operate in the "employment" mode or the "vertically integrated mode," it must control the transferable actions shown in row 3 of Table 1.

The main difference between the "employment" and "vertically integrated" modes is that, in the "employment" mode, the malware supplier employs the agent; whereas, in the "vertically integrated" mode, the malware supplier and the agent both work for a vertically integrated organization. If the malware supplier chooses to operate in the "platform" mode, it must enable the agent to control transferable actions.

What is less clear is how to best apply the general model with one agent developed by Hagiu and Wright (2015c) to the "reseller" and "input supplier" modes. Hagiu (2007) formally compared the "reseller" and "two-sided platform" modes using four fundamental economic factors: indirect network effects between buyers and sellers; asymmetric information between sellers and the intermediary; investment incentives; and product complementarities/substitutability. Hagiu concluded that the "reseller" mode is more profitable when the degree of complementarity among sellers' products is higher and it is very difficult to bring the two-sides to the platform together and spark interactions. The "two-sided platform" mode is preferred when seller investment incentives are important or when there is asymmetric information regarding seller product quality (Hagiu, 2007). This type of guideline focuses on constructs that are difficult to observe and would be difficult to apply in practice, particularly when studying the malware market.

Hagiu and Wright (2015a) compared the "multisided platform" mode with the "reseller" mode and concluded that the decision of which mode to select depends on whether suppliers affiliated to the platform or the reseller have more important information relevant to the optimal tailoring of marketing activities for each specific product. When applied to our example, we interpret the conclusion in Hagiu and Wright (2015a) to mean that the "reseller" mode requires the malware supplier to have control rights over important information that is relevant to assemble and update the product–market fit of the goods and services provided to customers.

The supplier input mode has not been formally studied as much as the other four modes have been. Hagiu and Wright (2015b) made two observations when informally comparing the "input supplier" and the "multisided platform" modes. The first observation was that, when a firm operates in the "input supplier" mode, not all relevant customer types are on board. However, when the firm operates in the "multisided platform" mode, all relevant customer types are affiliated to the platform. The second observation was that, when the firm operates in the "input supplier" mode, it does not benefit from indirect network effects between users and application developers.

For the purpose of our example, we interpret the observations by Hagiu and Wright (2015b) to mean that, when operating in the "input supplier" mode, the malware supplier derives benefits from bringing on board kit vendors as customers, but does not find significant benefits by bringing onboard the kit vendors' customers. We conclude that the malware supplier and the agent will invest in non-contractible actions related to supporting kit vendors but not downstream customers.

Contribution

The framework presented in this article can be used to anticipate the mode in which a malware supplier with one agent will function. If a malware supplier controls the un-contractible actions that could be carried out by the agent, it will function in the "employment" mode. If the malware supplier enables the affiliated agent to interact directly with affiliated customers, the malware supplier will function in the "multisided platform" mode. If the malware supplier has control rights over important information that is relevant to assemble and update product–market fit of the goods and services provided to customers, the malware supplier will operate in the "reseller" mode. If the malware supplier and the agent are both employed by the same organization, the malware supplier will function in the "vertically integrated" mode. If the malware supplier invests in non-contractible actions to support kit vendors but not downstream customers, it will operate in the "input supplier" mode.

The ability to anticipate the modes in which malware suppliers will function improves the classification of malware suppliers, agents, and customers; it enables defences to be tailored to address attacks of a particular type; it increases the number and quality of operational insights; it enables targeted operations; and it increases

Examining the Modes Malware Suppliers Use to Provide Goods and Services

Tony Bailetti and Mahmoud Gad

the productivity of experimenting with new ways of protecting organizations and individuals against cyberattacks.

The proposed framework can also be used to study which types of malware suppliers, agents, and customers are attracted to each mode, discover new modes, and specify the threat space a malware supplier poses. A better understanding of the actors that each mode attracts, an improved ability to discover new modes, and an improved specification of the threat space offers to lower the impact of improbable events such as those referred to as "black swan" events (Taleb, 2007).

Conclusions

We build on recent advances in the theory of multisided platforms to develop a framework that can be used to examine the various approaches that malware suppliers can take to deliver goods and services to customers. We provide an elemental model distilled from the general model with one agent developed by Hagiu and Wright (2015c). By elemental model, we mean that the model has been reduced to stark simplicity for the purpose of increasing its adoption as an integrative framework to formally examine the modes in which malware suppliers operate. This approach involves judgement, and it is consistent with research that attempts to formalize different theories (Gibbons, 2005). This elemental model is then used to identify five modes we believe that malware suppliers use to provide goods and services to their customers.

This study discusses the application of a theoretical model, essentially ignoring empirical testing and the formal mathematical proofs provided by the researchers to specify the various models. The next steps for this work are: i) to examine existing known marketplaces for the purpose of detailing the framework described in this article and ii) to develop a model with multiple agents and spillovers that is specific to the modes used by malware suppliers.

This article is the first step to develop a theoretically sound framework that can be used to examine the various modes that malware suppliers use to produce and sell malware.

We expect a more formal approach to characterizing the modes in which malware suppliers function will decrease the number and impact of cyberattacks.

About the Authors

Tony Bailetti is an Associate Professor in the Sprott School of Business and the Department of Systems and Computer Engineering at Carleton University, Ottawa, Canada. Professor Bailetti is the Director of Carleton University's Technology Innovation Management (TIM) program. His research, teaching, and community contributions support technology entrepreneurship, regional economic development, and international co-innovation.

Mahmoud M. Gad is a Research Associate at VENUS Cybersecurity. He holds a PhD in Electrical and Computer Engineering from the University of Ottawa in Canada and an MSc in Electrical and Computer Engineering from the University of Maryland in College Park, United States. His research interests include cybercrime markets, machine learning for intrusion detection, analysis of large-scale networks, and cognitive radio networks.

References

Ablon, L., Libicki, M. C. & Golay, A. A. 2014. *Markets for Cybercrime Tools and Stolen Data: Hackers' Bazaar*. Santa Monica, CA: Rand Corporation. http://www.rand.org/content/dam/rand/pubs/research_reports/RR600/RR610/RAND_RR610.pdf

Armin, J. & Komarov, A. 2013. *Mobile Fraud: Mobile Threats and the Underground Marketplace*. Lexington, MA: APWG. http://docs.apwg.org/reports/mobile/apwg_mobile_fraud_report_april_2013.pdf

Boudreau, K. 2010. Open Multisided Platform Strategies and Innovation: Granting Access vs. Devolving Control. *Management Science,* 56(10): 1849–1872. http://dx.doi.org/10.1287/mnsc.1100.1215

Boudreau, K. J., & Hagiu, A. 2009. Multisided Platform Rules: Multisided Platforms as Regulators. In A. Gawer (Ed.), *Multisided Platforms, Markets, and Innovation:* 163–191. Northampton, MA: Edward Elgar.

Choudary, S. P. 2015. *Platform Scale: How an Emerging Business Model Helps Startups Build Large Empires with Minimum Investment*. Platform Thinking Labs.

Clark, L. 2015. Hacker Forum Darkode Is Back and More Secure than Ever. *Wired,* July 28, 2015. Accessed February 1, 2016: http://www.wired.co.uk/news/archive/2015-07/28/darkode-back-and-more-secure

Edelman, E. 2015. How to Launch Your Digital Multisided Platform. *Harvard Business Review,* 93(4): 91–97.

Eisenmann, T., Parker, G., & Alstyne, M. V. 2006. Strategies for Two-Sided Markets. *Harvard Business Review,* 84(10): 92–101.

Examining the Modes Malware Suppliers Use to Provide Goods and Services

Tony Bailetti and Mahmoud Gad

Europol. 2015. Cybercriminal Darkode Forum Taken Down through Global Action. *Europol,* July 15, 2015. Accessed February 1, 2016: https://www.europol.europa.eu/content/cybercriminal-darkode-forum-taken-down-through-global-action

Gibbons, R. 2005. Four Formal(izable) Theories of the Firm? *Journal of Economic Behavior & Organization,* 58(2): 200–245. http://dx.doi.org/10.1016/j.jebo.2004.09.010

Goodin, D. 2014. Researchers Warn of New, Meaner Ransomware with Unbreakable Crypto. *Arstechnica,* January 6, 2014. Accessed February 1, 2016: http://arstechnica.com/security/2014/01/researchers-warn-of-new-meaner-ransomware-with-unbreakable-crypto/

Gu, L. 2013. *The Chinese Underground in 2013.* Irving, TX: Trend Micro. http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-the-chinese-underground-in-2013.pdf

Hagiu, A. 2007. Merchant or Two-Sided Platform? *Review of Network Economics,* 6(2): 115–133. http://dx.doi.org/10.2202/1446-9022.1113

Hagiu, A. 2009. Two-Sided Platforms: Product Variety and Pricing Structures. *Journal of Economics & Management Strategy,* 18(4): 1011–1043. http://dx.doi.org/10.1111/j.1530-9134.2009.00236.x

Hagiu, A. 2014. Strategic Decisions for Multisided Platforms. *MIT Sloan Management Review,* 55(2): 71–82.

Hagiu, A. & Wright, J. 2013. Do You Really Want to Be an eBay? *Harvard Business Review,* 91(3): 102–108.

Hagiu, A., & Wright, J. 2015a. Marketplace or Reseller? *Management Science,* 61(1): 184–203. http://dx.doi.org/10.1287/mnsc.2014.2042

Hagiu, A., & Wright, J. 2015b. Multi-Sided Platforms. *International Journal of Industrial Organization,* 43: 162–174. http://dx.doi.org/10.1016/j.ijindorg.2015.03.003

Hagiu, A., & Wright, J. 2015c. *Enabling Versus Controlling.* Harvard Business School: Working Paper, 16-002. Boston, MA: Harvard Business School.

Kamluk, V. 2009. *The Botnet Ecosystem.* Woburn, MA: Kaspersky Lab. http://latam.kaspersky.com/sites/default/files/knowledge-center/kl_botnet%20ecosystem.pdf

Kovacs, E. 2015. Hacking Forum Darkode Resurfaces. *Security Week,* July 28, 2015. Accessed February 1, 2016: http://www.securityweek.com/hacking-forum-darkode-resurfaces

Krebs, B. 2013. Styx Exploit Pack: Domo Arigato, PC Roboto. *Krebs on Security,* July 13, 2013. Accessed February 1, 2016: http://krebsonsecurity.com/2013/07/styx-exploit-pack-domo-arigato-pc-roboto/

Mathews, L. 2014. $100 Malware Kit Lets Anyone Build Their Own CryptoLocker. *Geek.com,* January 7, 2014. Accessed February 1, 2016: http://www.geek.com/apps/100-malware-kit-lets-anyone-build-their-own-cryptolocker-1581505/

Peters, S. 2015. Hacking Team 0-Day Shows Widespread Dangers of All Offense, No Defense. *DarkReading,* July 8, 2015. Accessed February 1, 2016: http://www.darkreading.com/attacks-breaches/hacking-team-0-day-shows-widespreaddangers-of-all-offense-no-defense/d/d-id/1321224

Samani, R. 2013. *Cybercrime Exposed: Cybercrime-as-a-Service.* Santa Clara, CA: McAfee. http://www.mcafee.com/ca/resources/white-papers/wp-cybercrime-exposed.pdf

Taleb, N. N. 2007. *The Black Swan: The Impact of the Highly Improbable.* New York: Random House.

The Chosunilbo. 2015. N. Korean Hackers Get Access to 'Unbeatable' Tools. *The Chosunilbo,* July 22, 2015. Accessed February 1, 2016: http://english.chosun.com/site/data/html_dir/2015/07/22/2015072201500.html

WikiLeaks. 2015. Hacking Team. *WikiLeaks,* July 8, 2015. Accessed February 1, 2016: https://wikileaks.org/hackingteam/emails/

Citation: Bailetti, T., & Gad, M. 2016. Examining the Modes Malware Suppliers Use to Provide Goods and Services. *Technology Innovation Management Review*, 6(2): 21–27. http://timreview.ca/article/965

Keywords: cybersecurity, cybercrime, malware, suppliers, modes, agents, customers, multisided platform

License Compliance in Open Source Cybersecurity Projects

Ahmed Shah, Selman Selman, and Ibrahim Abualhaol

> *"One bad apple can spoil the bunch."*
>
> Proverb

Developers of cybersecurity software often include and rely upon open source software packages in their commercial software products. Before open source code is absorbed into a proprietary product, developers must check the package license to see if the project is permissively licensed, thereby allowing for commercial-friendly inheritance and redistribution. However, there is a risk that the open source package license could be inaccurate due to being silently contaminated with restrictively licensed open source code that may prohibit the sale or confidentiality of commercial derivative work. Contamination of commercial products could lead to expensive remediation costs, damage to the company's reputation, and costly legal fees. In this article, we report on our preliminary analysis of more than 200 open source cybersecurity projects to identify the most frequently used license types and languages and to look for evidence of permissively licensed open source projects that are likely contaminated by restrictive licensed material (i.e., containing commercial-unfriendly code). Our analysis identified restrictive license contamination cases occurring in permissively licensed open source projects. Furthermore, we found a high proportion of code that lacked copyright attribution. We expect that the results of this study will: i) provide managers and developers with an understanding of how contamination can occur, ii) provide open source communities with an understanding on how they can better protect their intellectual property by including licenses and copyright information in their code, and ii) provide entrepreneurs with an understanding of the open source cybersecurity domain in terms of licensing and contamination and how they affect decisions about cybersecurity software architectures.

Introduction

There are many types of open source cybersecurity packages that developers can leverage for product development and include within their proprietary products. Examples include penetration testing software tools that assist with identifying vulnerabilities and intrusion detection tools that are used to detect cyber-attacks. However, whether or not an open source package can be included within a commercial product will depend on the package license and the extent to which it restricts commercial activities such as the sale of the software and keeping derivative code confidential.

For the purposes of this article, we divide licenses into two categories: permissive and restrictive. The permissive category includes commercial friendly licenses, such as BSD, Apache, and MIT. In contrast, the restrictive category includes comparatively commercial unfriendly licenses, such as the GPL, that restrict the sale of software that includes an open source package with such a license.

Intellectual property and legal compliance issues can arise when companies fail to implement a thorough license evaluation process when they consume open source. The challenge is accentuated by the absence of a forced to click "I agree" to the license terms before installing or using code (Gaff and Ploussios, 2012). Contamination could occur when restrictively licensed code is copied into a permissively licensed project package or when a restrictively licensed package is copied into a permissively licensed project. Developers that are working under tight deadlines can easily over-

License Compliance in Open Source Cybersecurity Projects

Ahmed Shah, Selman Selman, and Ibrahim Abualhaol

look the licensing commitments of what they consume unless they have policies and tools in place to prevent contamination (Khanafer, 2015). This ease of consumption increases the risk of contamination. Developers need to know whether they are consuming code that is permissively licensed (i.e., commercial friendly) or restrictively licensed (i.e., commercial unfriendly). In the simplest case, they can simply check by inspecting the package license (i.e., the "readme" file or the "LICENSE" file), but complications can arise if a permissively licensed project silently hides code licensed under an incompatible restrictive license. However, to reduce the risk of contamination, developers should not assume that the same degree of diligence has been undertaken by other developers who contributed code to another project that is now being consumed as part of a separate package. Projects must take care not to inherit the problems of others, which can spread across projects in a viral manner as the code is copied. In a 2007 survey by Saugatuck Technology, 21% of respondents felt security/open/community concerns could inhibit the adoption of open source, while 12% felt that licensing issues and risks were a concern (Cited in Hassin, 2007). Failure to adhere to the open source licensing terms can lead to costly litigation, damage to a company's reputation, and cost spent to remediate contaminated code. For example, in 2009, the Software Freedom Conservancy, Inc. brought legal action for copyright infringement against 14 commercial electronics distributors including Westinghouse Digital Electronics, Best Buy, and Samsung (Klasfelld, 2011). These companies distributed code from BusyBox (an open source tool) in their products without adhering to the BusyBox license. The license stated that inheritors of BusyBox must make their own source code available to the public. Thus, licensing and copyright violations, in many cases resulting from code contamination, are substantial issues affecting vendors of software that leverages open source projects.

Within the cybersecurity domain, we investigated the extent to which projects with permissive open source package licenses (i.e., LICENSE and README files that refer to Apache, BSD, or MIT) are contaminated with a restrictive licensed file (GPL v1.0, 2.0, 3.0 ext). We examined more than 200 open source cybersecurity projects as an initial, exploratory study. By studying code contamination in open source cybersecurity projects and providing related insights about how contamination can be avoided, we ultimately seek to help developers make clean and profitable products.

Our motivation for analyzing the cybersecurity open source domain over other open source software domains comes from the authors' cybersecurity research exploring what tools can be used to create cybersecurity products and what cybersecurity tools can support or differentiate non-cybersecurity software product offerings. In addition, cybersecurity tools are vastly varied in type and function. Such tools include cyber-threat intelligence-sharing tools, software-defined radio tools, vulnerability and exploitation tools, and anti-virus tools.

This article is divided into four sections. First, we review the literature on open source licensing; how open source licensing can influence architecture; and how restrictive license contamination can lead to litigation. Next, we outline the origins of the sample projects that were studied and the analytical methods used. Then, we present our results, including information on the license types found in the study; the coding languages used; how well intellectual property rights are claimed in the sample; cases where restrictive licenses contamination occurred in permissively licensed packages; and proposed areas for future work. We conclude with a summary of results and recommendations.

Literature Review

Restrictive licenses are generally considered "viral" because they require a consumer of the licensed code to distribute their own derivative source code under that same license. "Proprietary code distributed with or alongside GPL-licensed [open source software] as part of a larger program or application can in many cases be deemed a "covered work" along with the [open source software]. This means that the entire covered work – the proprietary code and OSS – can only be distributed under the GPL license terms" (Gaff and Ploussios, 2012). Permissive licenses allow consumers of the open source project to redistribute or sell the compiled binary without the need to expose any code to the public. Generally speaking, restrictive licenses allow consumers of the open source project to redistribute the compiled binary under the condition that the source code of the binary must be made available to the public and that the binary and source code cannot be sold. Not adhering to license terms in open source software could results in a copyright infringement claim or breach of contract, which may in turn lead to prohibition of further sales, impoundment and destruction of combined software, and legal fees (Gaff and Ploussios, 2012).

License Compliance in Open Source Cybersecurity Projects

Ahmed Shah, Selman Selman, and Ibrahim Abualhaol

Most licenses are reciprocal licenses meaning they force all derived works to be licensed under the same license associated with the original copy of the component (Link, 2011). The General Public License (GPL; gnu.org/licenses/gpl.html) is the most common and notable example. Permissive licenses such as MIT (opensource.org/licenses/MIT) and Apache (apache.org/licenses/) have fewer restrictions and generally do not require the user to distribute their own derived work. Due to the variation of terms in each license type, licenses can be incompatible with each other if they are within the same open source package. In other words, if a developer is considering using multiple types of licensed open source projects, there is a risk that the licenses will not be compatible and that software therefore cannot be combined (Lokhman et al., 2013). For instance, a package that is licensed with Apache 2.0 is not compatible with GPL 1.0. Therefore, GPL 1.0 code should not exist in the Apache-licensed package's code base. In this article, the terms “license conflict” or “contamination” refer to a project with a permissive license contains restrictively licensed code.

Open source licensing influencing architecture

We define derived work as the result of enhancing or editing open source software. Depending on developer intentions, either to distribute derived work or publish their work while maintaining copyright ownership, open source legality and licensing issues must be faced. One approach, used by the Linux kernel, is the "core-periphery pattern" (Lokhman et al., 2013). The core of the Linux kernel owns the copyright for the core system, while applications built around this system (i.e., on the periphery) can be replaced with different applications to allow any number of versions, or distributions, of Linux to be created for different purposes and systems. This approach allows for license-compatible customization, and thereby enables usability, scalability, and modularity.

The main problem facing commercial companies are the obligations associated with the derived work (Hammouda et al., 2010). First, they must be aware of the licenses of the different components used in their systems, and second, they must make sure all these licenses are compatible. However, in some cases, it is hard to find a suitable project that has the appropriately compatible licenses and, therefore, software architecture considerations arise.

Conflicts can prohibit the integration of open source components and require extra effort to understand the limitations of the licenses used (Link, 2011). Consider the difference between the Lesser General Public License (LGPL; gnu.org/copyleft/lesser.html) and the GPL license (gnu.org/licenses/gpl.html). For code under the LGPL, the user is permitted to link it dynamically to other components without violating or enforcing the LGPL (Lokhman et al., 2013). In contrast, this same scenario with the GPL requires a separate executable if the software code is not being released. Thus, this requirement of the GPL can affect the architecture of the entire system, particularly when there is a mix of proprietary and open source components. For example, instead of linking components with the GPL component through control-driven communication, data-driven relationships must be used instead (Hammouda et al., 2010). Another approach is to use the "isolation pattern", which separates components from each other to avoid license conflicts (Hammouda et al., 2010). Depending on the nature of the system (i.e., hosted, distributed, released as open source), the system architecture must appropriately accommodate licensing obligations.

Contamination leading to litigation

There are many ways that a company's product can end up containing restrictively licensed source code, potentially triggering GPL-related litigation. Common violations include not distributing the source code of derivative works or failing to add appropriate copyright information or licenses to derivative works.

Many GPL contamination cases that lead to litigation often go through the following process:

1. *Release:* a third-party developer creates original source that is released under GPL.

2. *Contamination:* a commercial entity "consumes" the GPL code and (knowingly or unknowingly) adds the code to their commercial product.

3. V*iolation:* the commercial entity releases their GPL-contaminated product while not adhering to GPL terms (i.e., they fail to make their own source code available to the public).

4. *Indictment:* a company takes legal action against the GPL violator for not complying with GPL terms.

5. *Resolution:* the outcome of litigation.

The outcomes of litigation can be substantial, including but not limited to:

- Reputational damage

License Compliance in Open Source Cybersecurity Projects

Ahmed Shah, Selman Selman, and Ibrahim Abualhaol

- Exposing customers to liability
- Threats of patent infringement for code tied to patents
- Making proprietary code open source
- Statutory damages
- Remediation costs of re-writing code

Research Method

The source code for over 200 cybersecurity projects were downloaded which included tools for penetration testing, forensic investigation, intrusion detection, visualization, and network monitoring. We developed our dataset of projects by sampling a subset of the tools listed from the following three primary sources:

1. Kali Linux OS distribution (Offensive Security, 2015)
2. Department of Homeland Security's list of open source cybersecurity software (DHS, 2012)
3. Security Onion Linux OS distribution (Burks, 2015)

An overall dataset of 334 open source cybersecurity projects was created and three levels of analysis was conducted:

1. *Attribution:* Across all 334 projects, we determined the extent of: i) copyright information in each file, ii) license information in each file, iii) no copyright or license attribution in each file. The purpose of this analysis was to determine how many files across all projects have no copyright or license attribution and also what types of license attribution were applied to a file, if any.
2. *License conflicts:* Out of the 334 open source cybersecurity tools that we downloaded, tools for which we could confirm the package license from the project's website or from the source code's package license (i.e., "COPYING" or "LICENSE" file) were selected for an analysis of license conflicts. The resulting subset of 255 projects were examined for evidence of GPL file contamination in a permissively licensed package. To look for patterns in the appearance of license conflicts, we evaluated license conflicts against the number of lines of code per package and the types and number of coding languages used in each of these projects.
3. *Third-party code:* Across a sample of 243 projects where we could confirm the licenses, we assessed the volume of third-party code as a proportion of the total code volume in each project. To look for patterns in the appearance of license conflicts, we evaluated license conflicts against the lines of code per package.

To conduct the three levels of analysis described above, we scanned and analyzed the downloaded software packages using Protecode's Enterprise System 4 code-scanning engine (protecode.com/our-products/system-4/). The analyses included determination of the number of lines of code per package; likely third-party volume per package; license type per package; programming languages used per package; if a copyright or license existed in a code file; and if a license conflict existed in a package. Protecode has a database containing millions of files from many open source projects hosted on several forges. When scanning the downloaded code, Protecode generates signatures and hashes that it compares against signatures of the files stored in Protecode's database. In this manner, Protecode's tools can identify if there are any matching files thereby indicating a file or part of the file exists in an open source project. Protecode also stores information regarding copyright and licenses of the open source projects found in the database, which will help identify any license conflicts between the open source components identified in the scanned code.

Results

Across the 255 projects where licenses could be determined, 24% had permissive licenses. Four packages out of the 61 permissively licensed projects were confirmed of being contaminated with GPL code. GPL contamination was confirmed by checking if the permissively licensed package contained a file with a GPL attribution (i.e., a GPL license within the file or a reference to a GPL license within the file). The cases of GPL contamination include permissively licensed packages that included one or more GPL licensed files or including whole GPL licensed packages (*.js, *.py, *.tar).

We also found other cases where GPL contamination might have occurred, but it could not be confirmed with high certainty. For example, two permissively licensed projects may have inherited GPL code (modified or un-modified) and the GPL code does not contain a GPL reference within it. In another case, we found information on the project website that claimed

License Compliance in Open Source Cybersecurity Projects

Ahmed Shah, Selman Selman, and Ibrahim Abualhaol

that a particular package was permissively licensed; however, when we downloaded the package, we found that the license was, in fact, restrictive.

Figure 1 compares the permissively licensed packages that are GPL contaminated with those that are not contaminated using a cluster plot of the total lines of unique code versus lines of third-party code. The figure shows that contaminated projects each have over 10,000 lines of third-party code and over 1,000 lines of unique code, although no other pattern is evident in this dataset, which contains only four cases of contamination.

Package licenses

Out of the 255 projects for which licensing information was available, 61 (24%) were found to have permissive licenses (i.e., MIT, BSD, or Apache). BSD-licensed projects were most common, accounting for nearly a third of permissively licensed projects and highlighting the flexibility inherited in this license. The MIT and Apache licenses were also common, each accounting for about 15% of the remaining permissively licensed projects. Of the permissively licensed packages contaminated with GPL-licensed code, two were licensed under BSD, one was licensed under MIT, and one contained a mix of permissive licenses.

The other 194 (76%) of 255 projects for which licensing information was available included a restrictively licensed (i.e., GPL) projects. One package was found to have a EUPL 1.1 license, which contained files that alluded to being GPL licensed. This package was grouped into the restrictive license category. Also included where packages were licensed under an LGPL (v3, v2, or v2.1), which could be considered moderately restrictive.

Figure 2 plots the number of programming languages used in each project against the number of lines of code for permissively and restrictively licensed packages. Figure 2 shows that packages with more than 1,000 lines of code are likely using one, two, or more languages, whereas packages with over 100,000 lines of code are likely using two or more programming languages. The GPL contamination boxes show the location of permissively licensed packages with GPL contamination. Out of the sample of 344 projects (which included projects that had licenses that could not be confirmed), the three most commonly used programming languages were C, Python, and PERL. This distribution of the four cases of GPL-code found in permissively licensed packages shows that contamination can occur regardless of the number of languages used.

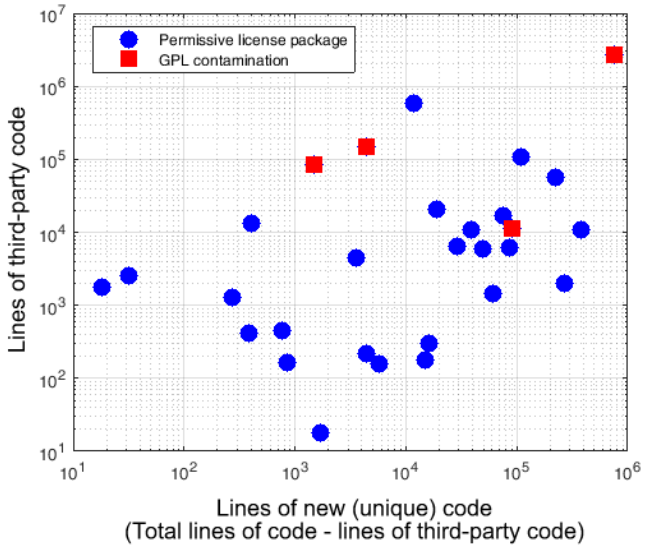

Figure 1. Cluster plot of projects with permisive package licenses by lines of unique code and likely third-party code

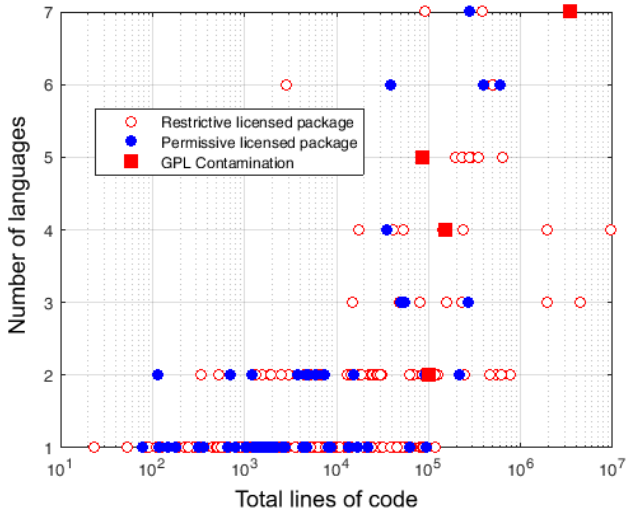

Figure 2. Programming languages versus total lines of code in restrictively licensed and permissively licensed packages, including evidence of GPL contamination in permissively licensed packages

License Compliance in Open Source Cybersecurity Projects

Ahmed Shah, Selman Selman, and Ibrahim Abualhaol

Copyright and license information in code files

Often, when open source code is brought into projects, what is inherited is not the entire package of another project, but only a code snippet or a file from that project. If intellectual property claims (copyright) or licenses are not embedded within the file, there is a risk that the file could be mistakenly be used in a permissively licensed project, and this mistake could then be propagated into other projects, leading to viral contamination. In our dataset of 334 packages, in which we found 151,187 files (not including binaries), 39% of the files had no copyright information or did not refer to a license. For the files that did have either copyright or license information, 2% percent only made reference to a license, 43% made reference to a license and contained copyright information, and 16% only had copyright information (Table 1). Out of the 45% that did refer to a license, 63% of the files made reference to GPL, and 13% were standalone (not mixed) Apache, BSD, or MIT licensed.

Volume of third-party code

Protecode Enterprise Server 4 was used to determine the amount of third-party code that likely exists in each project in our subsample of 243 projects. When the Protecode software scans a file, it compares it against its database of known third-party code. If the Protecode software provided a suggested best match of third party for a file, for the sake of this article, we treat the entire file as third-party code.

Figure 3 presents the distribution of lines of code across projects, highlighting the third-party code and also the permissively licensed packages that are contaminated with GPL material. Figure 4 shows the distribution of projects by the percentage of the code within the project that is likely from a third party. Around 145 projects contain 0% to 10% third-party code while around 20 projects contain 90% or more third-party code. Across all projects, the average volume of third-party code is 27%.

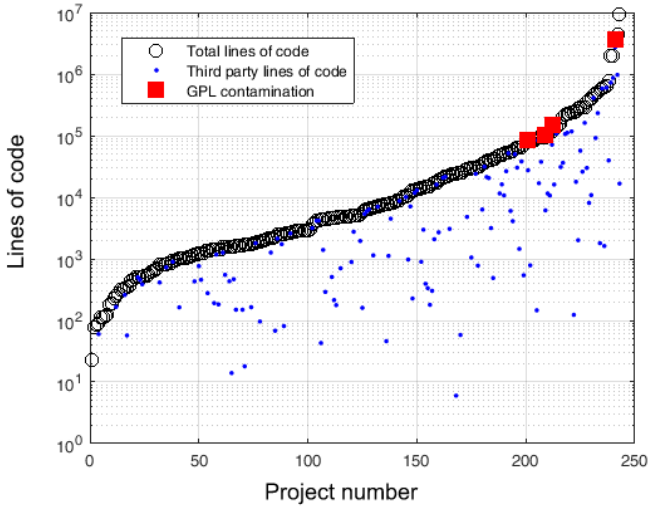

Figure 3. Extent of third-party code in the 243 sampled projects, ordered by total lines of code in each project

Table 1. Copyright and license information in 334 open source cybersecurity projects

<table>
<tr><td colspan="4">Full Dataset of 334 Projects:
151,187 files in total not including binaries</td></tr>
<tr><td colspan="3">IP or License Claim

Reference to license(s) or copyright

91,651 Files
61%</td><td>No IP or License Claim

No reference to copyright or license

59,536 Files
39%</td></tr>
<tr><td>Reference to license(s) only

3,267 Files
2%</td><td>Reference to license(s) and copyright

64,361 Files
43%</td><td>Reference to copyright only

24,023 Files
16%</td><td></td></tr>
</table>

License Compliance in Open Source Cybersecurity Projects

Ahmed Shah, Selman Selman, and Ibrahim Abualhaol

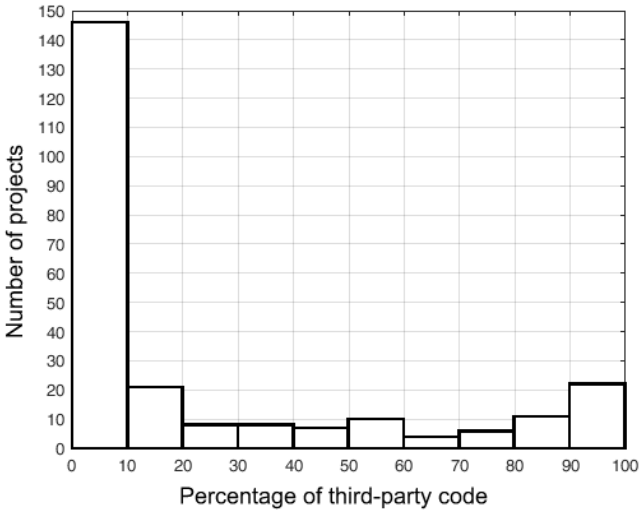

Figure 4. Histogram of number of projects versus the portion of package that is likely third-party code

Future Work

This article provided an initial, exploratory analysis of open source cybersecurity projects to provide insight on open source license conflicts. Our results provide developers with insights into the characteristics of open source cybersecurity projects in terms of lines of code, languages used, and license types. In addition, we tried to identify the risk of permissively license projects being contaminated with GPL and the extent to which developers are adding copyright and license references to their code.

Future work could include statistical correlation analysis between different attributes, investigating a greater number of attributes (e.g., the number of contributors), and analyzing more projects to increase the power of the analysis in terms of detecting or ruling out the appearance of cluster patterns. Such work could lead to a classification of contamination probabilities based on a k-Nearest Neighbour (KNN) Algorithm.

Conclusion

We found that the open source cybersecurity community is not adding copyright information or license references to files to claim intellectual property rights: 39% of files did not have copyright or license attribution. We suggest that managers should implement policies of adding copyright and licenses to their source code to ensure that intellectual property rights are claimed and to also make sure that GPL source code might not accidentally be consumed and contaminate a commercial product. We also found that there is no guarantee that packages with permissive licenses are not contaminated with restrictive licensed material: four out of 61 permissively licensed projects were contaminated with restrictive licenses. In addition, 76% of open source cybersecurity projects had restrictive package licenses and 24% had permissive package licenses. These findings suggest that the options for reusing open source code in the cybersecurity space are small with respect to selling proprietary software. However, the majority of restrictive licenses can be monetized through complementary services of open source products. Although much of the existing literature discusses the issue of open source licensing, licensing conflicts, and licensing compatibility, these studies are often light on data. In this study, we examined a dataset of over 300 open source cybersecurity projects and provides a stepping-stone for further investigation in the open source cybersecurity domain. Although our findings revealed only four cases of contamination across 344 open source cybersecurity projects, the potential ramifications of such contamination for those individual warrant further study into how companies can mitigate this risk.

License Compliance in Open Source Cybersecurity Projects

Ahmed Shah, Selman Selman, and Ibrahim Abualhaol

About the Authors

Ahmed Shah holds a BEng in Software Engineering and is pursuing an MASc degree in Technology Innovation Management at Carleton University in Ottawa, Canada. Ahmed has experience working in cybersecurity research with the VENUS Cybersecurity Corporation and has experience managing legal deliverables at IBM.

Selman Selman is a Software Engineer at Synopsys under the Software Integrity Group. He is also carrying out graduate studies in Technology Innovation Management at Carleton University in Ottawa, Canada.

Ibrahim Abualhaol holds BSc and MSc degrees in Electrical Engineering from Jordan University of Science and Technology, an MEng in Technology Innovation Management from Carleton University in Ottawa, Canada, and a PhD in Electrical Engineering from the University of Mississippi in Oxford, United States. He worked for two years as a Wireless Engineer at Broadcom Corporation and as a System Engineer Intern at Qualcomm Incorporation in the United States. He then worked as an Assistant Professor of Wireless Communications at Khalifa University, United Arab Emirates for four years. Currently, he is a Cybersecurity R & D Engineer working on operationalizing collective intelligence with artificial intelligence to improve cybersecurity. He is senior member of IEEE, a member of Phi Kappa Phi, and a member of Sigma Xi.

Citation: Shah, A., Selman, S., & Abualhaol, I. 2016. License Compliance in Open Source Cybersecurity Projects. *Technology Innovation Management Review*, 6(2): 28–35. http://timreview.ca/article/966

Keywords: cybersecurity, open source, license, copyright, GPL, third-party code, contamination

References

Burks, D. 2015. Security-Onion Project: Tools. *Security Onion Solutions*. Accessed November 1, 2015: https://github.com/Security-Onion-Solutions/security-onion/wiki/Tools

DHS. 2012. Open Source Cybersecurity Catalog: Homeland Open Security Technology (HOST) Project. *Department of Homeland Security (DHS) – Science and Technology Directorate*. Accessed November 1, 2015: https://www.dhs.gov/sites/default/files/publications/csd-host-open-soruce-cybersecurity-catalog.pdf

Gaff, B. M., & Ploussios, G. J. 2012. Open Source Software. *IEEE Computer Society*, 45(6): 9–11. http://dx.doi.org/10.1109/MC.2012.213

Hammouda, I., Mikkonen, T., Oksanen, V., & Jaaksi, A. 2010. Open Source Legality Patterns: Architectural Design Decisions Motivated by Legal Concerns. *Proceedings of the 14th International Academic MindTrek Conference:* 207–214. New York: ACM. http://dx.doi.org/10.1145/1930488.1930533

Hassin, K. 2007. Open Source on Trial. *Open Source Business Resource,* October 2007: 15–19. http://timreview.ca/article/66

Klasfelld, A. 2011. Westinghouse Sanctioned in Case Over Open Source. *Courthouse News Service:* August 12, 2011. Accessed February 1st, 2016: http://www.courthousenews.com/2011/08/12/38954.htm

Khanafer, H. 2015. Q&A. Does a Software Development Firm Need an Open Source Policy? *Technology Innovation Management Review*, 5(5): 45–46. http://timreview.ca/article/897

Link, C. 2010. Patterns for the Commercial Use of Open Source: Legal and Licensing Aspects. *Proceedings of the 15th European Conference on Pattern Languages of Programs (EuroPLoP '10):* Article No. 7. New York: ACM. http://dx.doi.org/10.1145/2328909.2328918

Lokhman, A., Mikkonen, T., Hammouda, I., Kazman, R., & Hong-Mei, C. 2013. A Core-Periphery-Legality Architectural Style for Open Source System Development. *Proceedings of the 46th Hawaii International Conference on System Sciences (HICSS):* 3148–3157. http://dx.doi.org/10.1109/HICSS.2013.34

Offensive Security. 2015. Index of Kali Linux Depot of Source Code. *Offensive Security*. Accessed July 2015: http://http.kali.org/kali/pool/main

TIM Lecture Series

Insights from Success and Failure in Technology Businesses

Chris McPhee, Peter Carbone, and Sean Silcoff

> *"If the rise and fall of BlackBerry teaches us anything, it is that the race of innovation has no finish line and winners and losers can change place in an instant. We live in an era of disruption where we are one algorithm away from being rendered redundant."*
>
> Sean Silcoff
> Business journalist and writer

Overview

The TIM Lecture Series is offered by the Technology Innovation Management (TIM; timprogram.ca) program at Carleton University in Ottawa, Canada. The lectures provide a forum to promote the transfer of knowledge between university research to technology company executives and entrepreneurs as well as research and development personnel. Readers are encouraged to share related insights or provide feedback on the presentation or the TIM Lecture Series, including recommendations of future speakers.

The first TIM lecture of 2016 was held at Carleton University on January 27th in celebration of the recent publication of the 100th issue of the *Technology Innovation Management Review* (timreview.ca/issue/2015/november). Chris McPhee, Editor-in-Chief of the TIM Review, and Tony Bailetti, Director of the TIM Program, invited two speakers – Peter Carbone and Sean Silcoff – to share lessons from studying key factors that have led to success and failure in technology businesses. This event also marked the launch of a new book comprising of the journal's 15 most popular articles as the latest installment in the Best of TIM Review book series (timbooks.ca).

Summary

Introduction: Chris McPhee
To set scene for the event, Chris McPhee described the evolution of the TIM Review and the recent publication of its 100th issue. This monthly, peer-reviewed journal has been published out of Carleton University's TIM program since 2007. It was first named the *Open Source Business Resource* and focused on the question of how businesses can make money by leveraging something that is free, namely open source software (McPhee, 2011). After publishing its first 50 issues, the journal was relaunched in October 2011 as the *Technology Innovation Management Review*.

The journal emphasizes research-based solutions to practical, real-world problems in emerging domains. It provides its authors and guest editors with opportunities to explore and legitimize new ideas. And, to encourage a diversity of perspectives – both in terms of having a global reach and receiving contributions from academia, industry, the public sector, etc. – the journal is open access and has no author charges (timreview.ca/authorguidelines).

In the 100th issue, McPhee (2015) contributed an article that looked back over the journal's first 100 issues, the themes they covered, trends in authorship and readership, and future opportunities and challenges for the journal. In its 100 issues, the journal has featured more than 500 articles by more 600 authors. The majority of authors have been from the Americas (55%) and Europe (40%), but the readership has been more internationally diverse, with the Americas accounting for 33%, Europe 30%, Asia 25%, Oceania 7%, and Africa 5%. The TIM Review website welcomes more than 27,000 unique visitors per month, with totals exceeding 600,000 unique visitors and 1,000,000 pageviews since the 2011 relaunch.

TIM Lecture Series – Insights from Success and Failure in Technology Businesses

Chris McPhee, Peter Carbone, and Sean Silcoff

With no author charges and a full open-access model (all articles are free to readers), McPhee explained that funding must come from other sources. A recent initiative is the development of a "Best of TIM Review" ebook series, and the seventh book in the series was launched during the TIM Lecture. The proceeds from every book sold go towards the ongoing operation of the journal.

This new book – titled *Best of TIM Review: Most Popular Articles* (McPhee, 2016; amazon.ca/dp/B01AZW6J98/) – features the 15 most popular articles published in the TIM Review based on visits to the website. It provides valuable insights on fostering entrepreneurship, managing innovation and teams, and delivering value to customers, and it will be of interest to entrepreneurs, managers, researchers, and others.

The TIM Books website (timbooks.ca) provides details on the entire TIM Books series, which includes the following titles:

1. *For Technology Entrepreneurs*

2. *Business Models for Entrepreneurs and Startups*

3. *Value Co-Creation*

4. *Cybersecurity*

5. *Open Source for Entrepreneurs*

6. *Living Labs*

7. *Most Popular Articles*

Part I: Peter Carbone – Lessons from the Evolution of Business

In inviting a speaker to mark the occasion of the 100th issue of the TIM Review, Peter Carbone was the obvious choice given his many contributions dating back to his authorship of the first article ever published in the journal (Carbone, 2007) and several subsequent contributions, including an article that appears in the new ebook of the journal's most popular articles. He has also contributed as a top-quality guest editor, reviewer, and advisory board member, and he has been a key factor in the journal's success. In the first part of the TIM Lecture, Carbone shared the lessons he has learned about the evolution of business as a technology executive and practitioner of strategic planning, and he mapped this evolution to the corresponding changes seen in the topics explored in the TIM Review over its first 100 issues.

Carbone explained that, over the last 30 to 40 years, there have been enormous shifts in virtually every factor that affects a successful business, and, for much of this time, industry has worked with academia to try to understand the levers to gain competitive advantage, or to even just keep up in the market. Based on examples from his experiences with this evolution in the telecommunications industry, Carbone identified seven areas of insight and key lessons learned, which are summarized below:

1. Customers

- Sales is not about the product; it is about putting the customer's well-being ahead of your own. Engage with customers, and make sure they are always at the table and part of the solution.

- Long-term planning is strategically critical for any company. It helps you decide what to do, but also what not to do and with whom to partner.

- You have to understand the physics of a deal. Is the customer right? What are they really asking for? Do they really know what they need or what technology to bet on? Do your homework and have your own opinion.

2. Competition

- There is always competition, and it increasingly coming from unique and disruptive combinations of fundamental elements.

- Business model disruption is hard to anticipate, address, or leverage.

- The winner is the one who appropriates value, not necessarily the one who creates it. And, there are different ways to appropriate value.

3. Open Source

- Open source is a valuable strategic tool, not just an engineering tool.

- It can be used to take out competition by reducing their strategic advantage, thereby shifting the value to where a company can compete.

- Open source facilitates collaborative innovation, but there are cultural implications: developers and managers must overcome the perceived need to own and

TIM Lecture Series – Insights from Success and Failure in Technology Businesses

Chris McPhee, Peter Carbone, and Sean Silcoff

control everything and instead learn to trust and work with a community.

4. Business ecosystems

- An ecosystem can magnify capability, enhance reach, and improve responsiveness. The benefits of a successful ecosystem are substantial, but setting one up is easier said than done, largely because of trust issues.
- Particularly for custom development, a business ecosystem can be a mechanism for partnership with customers.
- Success depends on the ecosystem providing a "win-win" for everyone in terms of commercial success. Participants must have mutual self-interest.

5. Mergers and acquisitions

- Mergers and acquisitions are a popular mechanism to address time to market, to contain investment, to enter new markets, and for the scaling or exit of startups.
- There are various possible models of integration: the right choice depends on what is motivating the merger or acquisition. For details, see Carbone (2011).

6. Investments in the future

- Large companies manage investment across three horizons: immediate impact (H1), short-term impact (H2), and long-term impact (H3). The challenge is to balance investment across all three horizons without sacrificing one at the expense of another. This task is not easy because each horizon has different, and often competing, characteristics.

7. People

- A CEO requires deep knowledge of the business but cannot succeed if their influence clashes with the company culture; they must command respect and be followed. Trust and respect drive productivity.
- Empowering means incenting, but you must create a specific incentive to achieve a particular outcome.
- A company is a complex community that should be leveraged, not overridden.

Part II: Sean Silcoff – Lessons from the Rise and Fall of BlackBerry

Next, Sean Silcoff provided an inside look at the rise and fall of one of Canada's most iconic brands, as documented in his recent book *Losing the Signal* (McNish & Silcoff, 2015). Silcoff spoke about BlackBerry under the leadership of former CEOs Jim Balsillie and Mike Lazaridis. The book, co-written with Jacquie McNish, grew from a major feature investigation published in September 2013 in the Globe and Mail about the downfall of the Canadian company (Silcoff et al., 2013).

Silcoff shared profiles of the two former co-CEOs, their personalities and strengths and weaknesses, and how they were both mismatched and complementary within their unusual joint leadership of Research in Motion (now BlackBerry). He characterized the company's rapid growth and how it established a dominant position in a market it created only to see their advantage slip away as Apple and others disrupted and redefined the smartphone industry.

Key lessons Silcoff shared from the investigation and book included:

1. Timing is everything: you need the right product in the right place at the right time.

2. Conviction is very important: you cannot always listen to what customers think they need.

3. Innovation without commercialization is not enough: you need to get the technology into people's hands.

4. Leadership is key: you can have co-CEOs, but any board that allows such a structure must regularly revisit the arrangement, possibly as often as every board meeting.

5. Technology is only part of the disruption story: it is just as important to change the rules to destabilize legacy businesses.

6. With the current rates of innovation, time is a luxury that you cannot afford.

TIM Lecture Series – Insights from Success and Failure in Technology Businesses

Chris McPhee, Peter Carbone, and Sean Silcoff

About the Speakers

Chris McPhee is Editor-in-Chief of the *Technology Innovation Management Review.* Chris holds an MASc degree in Technology Innovation Management from Carleton University in Ottawa, Canada, and BScH and MSc degrees in Biology from Queen's University in Kingston, Canada. He has over 15 years of management, design, and content-development experience in Canada and Scotland, primarily in the science, health, and education sectors. As an advisor and editor, he helps entrepreneurs, executives, and researchers develop and express their ideas.

Peter Carbone is a successful executive known for his thought leadership, business acumen, and technology leadership. He is often called on to address new business and technology challenges. Peter is a pathfinder with a track record of creating innovative solutions, strategically managing technology and innovation, successfully launching and running new businesses, and leading business development initiatives. Peter has held CTO, R&D, and senior business positions in several high-tech companies, and he has led or been directly involved with several technology company acquisitions. Peter has been engaged as technical advisor to startups, is part of the faculty of an entrepreneur development program that has created >100 new companies, and has been on the boards of US-based Alliance for Telecommunications Industry Solutions (ATIS) and a not-for-profit economic development company. He is past Vice-Chair of the Executive Committee of the Information Technology Association of Canada (ITAC) and Chair of an ITAC committee, which is focused on the Global Competitiveness of Canada's Knowledge Economy. Peter is also a member of the Advisory Board and Review Board of the *Technology Innovation Management Review.*

Citation: McPhee, C., Carbone, P., & Silcoff, S. 2016. TIM Lecture Series – Insights from Success and Failure in Technology Businesses. *Technology Innovation Management Review,* 6(2): 36–39. http://timreview.ca/article/967

Keywords: Technology Innovation Management Review, TIM Review, technology, innovation, management, lessons, insights, Nortel, Blackberry, Research in Motion, book launch

Sean Silcoff is co-author of Losing the Signal and a business writer with *The Globe & Mail,* Canada's National Newspaper. During his 21-year career in journalism and communications, he has covered just about every area of business, from agriculture to the credit crisis, toys to airplane manufacturing and steel to startups. He previously worked at the *National Post* as well as *Canadian Business Magazine,* where he oversaw publication of the inaugural edition of the Rich 100, the magazine's annual survey of Canada's wealthiest people. Sean is a two-time winner of the National Newspaper Award, the Montreal Economic Institute Economic Education Prize and the Hon. Edward Goff Penny Memorial Prize for Young Canadian Journalists. He led The Globe & Mail's coverage of the fall of BlackBerry. Sean has a business degree from Queen's University in Kingston, Canada, and a journalism degree from Carleton University in Ottawa, Canada.

This report was written by Chris McPhee.

References

Carbone, P. 2007. Competitive Open Source. *Open Source Business Resource,* July 2007: 4–6. http://timreview.ca/article/93

Carbone, P. 2011. Acquisition Integration Models: How Large Companies Successfully Integrate Startups. *Technology Innovation Management Review,* 1(1): 26–31. http://timreview.ca/article/490

McNish, J., & Silcoff, S. 2015. *Losing the Signal: The Spectacular Rise and Fall of BlackBerry.* Toronto: Harper Collins. http://harpercollins.ca/9781443436182/

McPhee, C. 2011. Reflecting on Fifty Issues of the OSBR. *Open Source Business Resource,* August 2011: 32–36. http://timreview.ca/article/465

McPhee, C. 2015. Reflecting on 100 Issues of the TIM Review. *Technology Innovation Management Review,* 5(11): 5–11. http://timreview.ca/article/940

McPhee, C. (Ed.) 2016. *Most Popular Articles: Best of TIM Review.* Ottawa: Talent First Network (Carleton University). http://amazon.ca/dp/B01AZW6J98/

Silcoff, S., McNish, J., & Ladurantaye, S. 2013. Inside the Fall of BlackBerry: How the Smartphone Inventor Failed to Adapt. *The Globe and Mail,* September 27, 2013. Accessed February 15th, 2016: http://www.theglobeandmail.com/report-on-business/the-inside-story-of-why-blackberry-is-failing/article14563602/

Author Guidelines

These guidelines should assist in the process of translating your expertise into a focused article that adds to the knowledge resources available through the *Technology Innovation Management Review*. Prior to writing an article, we recommend that you contact the Editor to discuss your article topic, the author guidelines, upcoming editorial themes, and the submission process: timreview.ca/contact

Topic

Start by asking yourself:

- Does my research or experience provide any new insights or perspectives?
- Do I often find myself having to explain this topic when I meet people as they are unaware of its relevance?
- Do I believe that I could have saved myself time, money, and frustration if someone had explained to me the issues surrounding this topic?
- Am I constantly correcting misconceptions regarding this topic?
- Am I considered to be an expert in this field? For example, do I present my research or experience at conferences?

If your answer is "yes" to any of these questions, your topic is likely of interest to readers of the TIM Review.

When writing your article, keep the following points in mind:

- Emphasize the practical application of your insights or research.
- Thoroughly examine the topic; don't leave the reader wishing for more.
- Know your central theme and stick to it.
- Demonstrate your depth of understanding for the topic, and that you have considered its benefits, possible outcomes, and applicability.
- Write in a formal, analytical style. Third-person voice is recommended; first-person voice may also be acceptable depending on the perspective of your article.

Format

1. Use an article template: .doc .odt
2. Indicate if your submission has been previously published elsewhere. This is to ensure that we don't infringe upon another publisher's copyright policy.
3. Do not send articles shorter than 1500 words or longer than 3000 words.
4. Begin with a thought-provoking quotation that matches the spirit of the article. Research the source of your quotation in order to provide proper attribution.
5. Include a 2-3 paragraph abstract that provides the key messages you will be presenting in the article.
6. Provide a 2-3 paragraph conclusion that summarizes the article's main points and leaves the reader with the most important messages.
7. Include a 75-150 word biography.
8. List the references at the end of the article.
9. If there are any texts that would be of particular interest to readers, include their full title and URL in a "Recommended Reading" section.
10. Include 5 keywords for the article's metadata to assist search engines in finding your article.
11. Include any figures at the appropriate locations in the article, but also send separate graphic files at maximum resolution available for each figure.

Issue Sponsor