Philosophy & Technology
https://doi.org/10.1007/s13347-018-0329-z
RESEARCH ARTICLE

Building General Knowledge of Mechanisms in Information Security

Jonathan M. Spring[1] · **Phyllis Illari**[2]

Received: 22 June 2018 / Accepted: 14 August 2018 / Published online: 17 September 2018

Abstract

We show how more general knowledge can be built in information security, by the building of knowledge of mechanism clusters, some of which are multifield. By doing this, we address in a novel way the longstanding philosophical problem of how, if at all, we come to have knowledge that is in any way general, when we seem to be confined to particular experiences. We also address the issue of building knowledge of mechanisms by studying an area that is new to the mechanisms literature: the methods of what we shall call mechanism discovery in information security. This domain offers a fascinating novel constellation of challenges for building more general knowledge. Specifically, the building of stable communicable mechanistic knowledge is impeded by the inherent changeability of software, which is deployed by malicious actors constantly changing how their software attacks, and also by an ineliminable secrecy concerning the details of attacks not just by attackers (black hats), but also by information security defenders (white hats) as they protect their methods from both attackers and commercial competitors. We draw out ideas from the work of the mechanists Darden, Craver, and Glennan to yield an approach to how general knowledge of mechanisms can be painstakingly built. We then use three related examples of active research problems from information security (botnets, computer network attacks, and malware analysis) to develop philosophical thinking about building general knowledge using mechanisms, and also apply this to develop insights for information security. We show that further study would be instructive both for practitioners (who might welcome the help in conceptualizing what they do) and for philosophers (who will find novel insights into building general knowledge of a highly changeable domain that has been neglected within philosophy of science).

Keywords Mechanistic explanation · General knowledge · Multifield mechanisms · Building mechanistic knowledge · Computer security incident response

✉ Jonathan M. Spring
jspring@cs.ucl.ac.uk

Extended author information available on the last page of the article.

1 Introduction

Scientists from many disciplines explain phenomena mechanistically. Different accounts of mechanistic explanation have been offered in the philosophy of science literature, leading to the emergence of something like a core consensus view referred to as "minimal mechanism" in Glennan and Illari (2017): "A mechanism for a phenomenon consists of entities (or parts) whose activities and interactions are organized so as to be responsible for the phenomenon."[1]

Within philosophy of science the mechanisms literature actually exists as two largely parallel literatures, one studying mechanistic explanation in the life sciences, broadly construed (Bechtel and Richardson 1993; Machamer et al. 2000; Glennan 2005), while the other studies the social sciences (Elster 1983; Steel 2008; Kincaid 2011). Here, we will begin to explore how to extend this interesting work to a major branch of science that has been largely neglected by the mechanisms literature: computer science.[2]

There are some exceptions to the general neglect. Such papers as Piccinini (2007) examine what it means for a process or mechanism to be characterized as a computation, Floridi et al. (2015) consider the notion of malfunction of a computation, and Angius and Tamburrini (2017) discuss explanation of the behavior of computing systems. The impact of information security practices on discovery in medicine is discussed by Tempini and Leonelli (2018), but the focus is data curation in medicine. Galison (2012) has broadly stated knowledge in "Manichaean Sciences" such as cybersecurity should be local and intercalated, and result from trading zones. Quite different from these, our focus will be how computer scientists build more general mechanistic knowledge, in a way that we shall describe.

For concreteness, we restrict our work on mechanisms to examples from information security (henceforth "infosec"),[3] which is the subdiscipline of computer science concerned with ensuring simultaneously the confidentiality, integrity, and availability of IT systems against the attacks of some set of adversaries.[4] Our goal is to show

[1] Compare (Illari and Williamson 2012; Craver 2007; Glennan 2017).

[2] There has been a heated debate on the status of computer science as a science. Tedre (2011) concludes in his survey of the discussion of the disciplinary status of computing by stating "there is nothing wrong either in considering computing to be a science, or considering it to be something else. Each of those arguments just assumes a specific viewpoint about both science and computing" (p. 382). Similarly, Tedre and Moisseinen (2014, p. 8) argue from their survey on experimentation in computing that philosophy of science should attempt to understand computing in its own right, rather than inflict idealized views on scientific methods from other fields on computing. Spring et al. (2017) argue specifically that information security is a science from a critical survey of the science of security literature. We follow the conclusions of these surveys and treat computer science, and information security in particular as a subfield, as scientific disciplines from which philosophy of science can draw novel insights.

[3] The field may variably be called information security, computer security, cybersecurity, or even the laborious "information and computer technology resiliency management." We use "information security," shortened to "infosec," to emphasize that human users and the physical world are part of the sociotechnical system under study, alongside computers or cyberspace.

[4] Information security works entirely in the space between confidentiality and integrity on the one hand and availability on the other. One common quip is that the best confidentiality and integrity protection for your computer is to turn it off, unplug all the wires, encase it in concrete, and drop it to the bottom of the ocean. Availability is the necessary counterbalance.

how general knowledge can be built even in infosec. We show this using three case studies (which we will show are interrelated in interesting ways). Three cases cannot illustrate how knowledge is built in all of infosec, much less all of computer science. Indeed, there may be different ways of building more general knowledge, and there may be cases which are too difficult to build anything very general. Nevertheless, there are three important gains to the philosophical literature. We profile cases where general knowledge is built via methods wildly different from the majority of cases studied to establish previous philosophical theories of knowledge. We demonstrate that in this unstudied domain, knowledge can be built in a way that is not well accounted for in existing philosophical approaches to knowledge. And we show that the mechanisms literature provides positive insights into how general knowledge is built in these difficult cases. Whether the mechanisms approach will generalize to further cases in computing is an area for future work.

Infosec faces a novel constellation of challenges that make it particularly worth philosophical study. We have selected three interlocking challenges with significant impact on both research and methodology. First, the immediate object of study, namely software, can change behavior during observations or between them; second, practitioners face active adversaries that respond to, and try to confound, observations; and, third, secrecy even among friendly practitioners is common because successful strategies need to be hidden from attackers, and participants need to protect their other interests.[5]

Computer science has a theoretical basis, and the logical basis of computers and computation may suggest that there should be a logical, a priori answer as to what a program does and whether that behavior is secure. However, this is true neither in principle nor in practice. Turing (1936) famously proved that it is in principle not possible to calculate a priori whether a computer program will halt.[6] If one cannot determine if a program will halt, then one cannot determine how many resources it will use, and therefore how many resources to allocate to it. There are many exhaustible resources, including RAM, processor cycles, disk space, processor thread identifiers, and file system identifiers. If a computer runs out of a resource, at best it stops responding to new requests. At worst, a computer may run out of the resources to remain stable and crash in a way that makes it vulnerable to adversary attacks.

For extremely small systems, in-practice heuristics and cautious overestimation can overcome this principled hurdle. However, any computer system actually in use has a complex supply chain for both hardware and software that cannot be considered a small system. Furthermore, security in-practice is a risk assessment: balancing costs and benefits, and balancing availability with confidentiality and integrity. Appetite for risk impacts productivity and profits; there is no uniquely correct answer to how much risk is the correct amount. The goal of infosec is merely to find a satisfactory

[5]This list is not exhaustive. Additional challenges which we will not discuss include the practical difficulty of detecting rare events while suppressing alarm errors (Axelsson 2000), economic incentives that work against secure systems (Anderson 2001), and navigating a changing international legal landscape.

[6]This famous "halting problem" is closely related to Church's Thesis on what mathematical functions are calculable and to Gödel's incompleteness theorem (Boolos et al. 2002). There is also a subfield of computing, complexity theory, dedicated to determining equivalence classes of how difficult a computable result is to derive in practice.

level of risk (of failure), both in a defensive posture and in the design of observations or experiments to detect adversaries.

Instead of building scientific theories, infosec practitioners model modes of attack and of defense in ways that we will show can usefully be thought of as modeling mechanisms of attack and of defense.[7] Building general knowledge in domains of this kind does not come by derivation from theory, as will gradually become clear in Sections 2 and 3. Therefore, we will turn to infosec practice in accord with the Philosophy of Science in Practice approach (SPSP 2017), rather than attempting to focus exclusively on theories, as has been more common in the history of philosophy of science. Indeed, rather than assuming general knowledge is there to find, we are interested in how difficult it is to build.

Our argument is organized in three parts. In Section 2, we briefly examine how the challenge of gaining general knowledge has been treated in philosophy of science. In Section 3, we explain the practice of building what we treat as knowledge of mechanisms in infosec. We begin in subsection 3.1 by developing the three major challenges of infosec we have briefly mentioned above. Then, in subsection 3.2, we begin our detailed casework with three examples. The first is the "intrusion kill chain" (Hutchins et al. 2011), a common model of the categories of steps an adversary must take in order to penetrate and control a system. At one level below the kill chain, practitioners analyze the malicious software (or malware) that accomplishes a particular step in the chain. This task is called "malware reverse engineering." At one level up from the intrusion kill chain mechanism, we use the UK's National Crime Agency's (NCA) model of the mechanism of money theft and laundering by the internet criminal ecosystem (Addis and Garrick 2014). We use these three examples of discovery to show how infosec work in these areas can broadly be thought of as the discovery and modeling of three interrelated mechanisms.

We use this detailed casework in Section 4 to show how fragmented work can still be seen as building, in a patchwork way, considerably more general knowledge even in the face of the three infosec challenges. Section 5 summarizes the positive impacts of understanding general knowledge as built up by discovering and modeling clusters of multifield mechanism schemas related along four dimensions.

2 Generality in Philosophy of Science

Approaches to general knowledge in the history of philosophy of science were for a long time focused on scientific theory and laws. We will examine that history in Section 2.1, showing that some initial discussion of knowledge in infosec has set off

[7]Note that we will not address metaphysical issues at all in this paper, although one of the authors has done this elsewhere (Illari and Williamson 2013). Epistemology is our primary concern. We will, however write about both mechanisms and models of mechanisms. There are some debates that might suggest this is controversial, particularly the ontic-epistemic debate, also addressed by one of the authors (Illari 2013). We take our work here to be in accord with the views even of the most ontic of major mechanists, in particular Craver (2006) and Glennan (2017), who, in discussing modeling extensively, both recognize the epistemic aspects of mechanism discovery far more than is usually recognized in discussions of their work.

on a familiar treacherous path, talking of scientific laws, and discussing whether or not there is a science of security on that basis. We follow the argument by Spring et al. (2017) that infosec should follow lessons from the more recent philosophy of the life and social sciences. To develop this recommended path, Section 2.2 extracts from the philosophical mechanisms literature a coherent thread of how mechanism discovery builds general knowledge in the life sciences.

Note that we will not directly address philosophical views of knowledge and generality, instead studying how the focus of such work has moved away from laws and theories as primary scientific knowledge, and this carries with it changes in how scientific knowledge is and should be seen. For those familiar with the philosophical literature, we endorse Hans Radder's 2017 survey of the ways in which a justified true belief account of knowledge is not fitting for scientific knowledge. Instead, we take a broadly pragmatist approach to knowledge and understanding, current proponents of which include Leonelli (2009) and Radder (2017).

2.1 Turning Away from Laws

There has been a longstanding philosophical interest in understanding what unified or general knowledge is, and how to build it. Unity of knowledge was traditionally understood in terms of unifying theory, driven by important theoretical unifications in science. A representative description of unification is given by Bogen and Woodward (1988, p. 325), when it was already being criticized:

> A characteristic kind of advance in scientific understanding occurs when one sees how what previously seemed to be a number of independent, unrelated facts can be accounted for in terms of a small set of common mechanisms or laws. Nineteenth-century optical theories represented an important explanatory achievement because they provided a unified, systematic account of a wide range of optical phenomena involving reflection, refraction, diffraction, stellar aberration, and polarization in terms of a few basic assumptions regarding the transverse wave character of light. Similarly, Maxwell's theory provided a unified treatment of an apparently diverse set of electromagnetic phenomena.

Bogen and Woodward (1988) already give one influential argument for the view that focusing exclusively on theory and laws was not adequate. With the growth of philosophy of life sciences, this questioning accelerated. Philosophers noticed that general knowledge in the life sciences cannot be understood in this way. There are relatively few unifying theories in the life sciences, and very few laws in the traditional sense. One possible exception is evolutionary theory and its various mathematical expressions, but even this, on its own, is not going to be adequate to capture everything that is known. Many philosophers of life sciences rejected a philosophy of science based on traditional laws, following Cartwright (1983), recognizing the plurality and diversity of the life sciences that makes laws rare (Mitchell 2003; Dupré 2012).[8]

[8]Some philosophers sought to re-characterize laws to be friendlier to the life sciences, such as Mitchell's pragmatic laws (Mitchell 1997), or Woodward's invariant generalizations (Woodward 2003). As our focus in this paper is mechanisms, we do not discuss these.

Infosec seems to face a similar problem to the life sciences in understanding what counts as general knowledge and how to build it. This challenge has likewise manifested as a kind of wrestling with the problem of laws. The US Department of Defense (DoD) is a major funder of security research, and as such has extensive influence over what is considered scientific in the field. In 2010, the DoD commissioned a study which posed for itself the following questions:

> Are there "laws of nature" in cyberspace that can form the basis of scientific inquiry in the field of cybersecurity? Are there mathematical abstractions or theoretical constructs that should be considered?

And answered with:

> There are no intrinsic "laws of nature" for cybersecurity as there are, for example, in physics, chemistry or biology. Cybersecurity is essentially an applied science that is informed by the mathematical constructs of computer science such as theory of automata, complexity, and mathematical logic (JASON Office 2010, p. 4).[9]

For a fuller discussion of uses of "laws" in contemporary questions about a science of infosec, see Herley and van Oorschot (2017) and Spring et al. (2017). So it seems there are parallel problems of understanding general knowledge in the life sciences and in infosec. This means we should be able to use work on the life sciences to help address infosec challenges. In philosophy of life sciences, attention has turned away from laws, to the search for mechanisms. However, the initial focus of the mechanisms literature was to give an account of scientific explanation without using laws, which means there has not been a lot of work directly on the question of building general knowledge by discovering mechanisms. There has been work on singular versus general mechanisms (Glennan 1997; 2011), on how we should think about really fragile mechanisms (Glennan 2010) and on regularity (Andersen 2017), but comparatively little on how general knowledge is built, analogous to the idea of theory-building. This means that, in philosophy of science, and in spite of the turn away from laws in philosophy of life sciences, current efforts to understand unification or generality are still largely dependent on this history of focusing on laws or arguments, developing the traditional Hempelian framework (Hempel 1965; Friedman 1974; Kitcher 1981). So, an important contribution of this paper is to use study of infosec to develop the philosophical literature on mechanisms and generality.

[9]The US DoD is influential worldwide on these issues. For example, the Canadian view approvingly quotes the US view about whether there are laws of nature: "No, since cybersecurity is an applied science informed by the mathematical constructs of computer science such as automata theory, complexity, and mathematical logic" (CSEC 2013). On a slightly different line, a main goal of the GCHQ-funded Research Institute in Science of Cyber Security is "to move security from common, established practice to an evidence base, the same way it happened in medicine" (University College London 2017).

2.2 Generality and Mechanisms

Our focus in this paper is how general knowledge can be built in infosec, and we have explained why we think the philosophical mechanisms literature is a more promising way to approach this than thinking in terms of laws. However, we still need an account of building of more general mechanistic knowledge to help us with infosec.

In this section, we will weave together Darden's work on clusters of mechanism schemas, Craver's discussion of the way multifield mechanisms can form what he calls a "mosaic unity," and Glennan's very recent work on the dimensions of variation of mechanisms. We will show that threads can be pulled out of this work and can be woven into a picture of general mechanistic knowledge being painstakingly built as some mechanisms become well known, alongside related ones, while various interesting relationships among those mechanisms gradually become better established. The picture that emerges is rather than generality being something given with laws or theory, it is something painstakingly built up in the mechanism discovery process. In line with this, generality is something much less than universal, which was the original assumption which went hand-in-hand with the study of supposedly universal laws. Generality turns out to be hard to find, and highly valued when it is.

Lindley Darden (2006) suggests that if there is such a thing as biological theory, it consists of clusters of mechanism schemas. Mechanism schemas are, in crude terms, abstractly specified mechanisms, lacking concrete detail. Mechanism schemas apply to far more things, such as cells and kinds of organisms, than concretely specified mechanisms, which always include detail that is particular to specific situations.[10]

For example, protein synthesis can be described at an extremely abstract level as the process by which genetic material is used by living things to create the proteins essential for life. This is often understood as involving two paradigm mechanism schemas: one for cells with a nucleus and one for cells without, each transcribing DNA to mRNA, which then moves to ribosomes to translate the mRNA code into amino acid chains. These are slightly different schemas, but each schema applies to many different kinds of cells, so each captures something quite general about protein synthesis. Together, knowledge of both schemas captures something even more general about protein synthesis—which includes the divergence between eukaryotes and prokaryotes. Going further, one more thing we know about protein synthesis is that lots of organisms use non-standard methods. One important example is protein synthesis as performed by HIV. HIV holds its genetic material as RNA, which it reverse transcribes into DNA, inserting that DNA into the genome of its host cell, to borrow the protein synthesis apparatus of the cell. But what was first discovered for a particular virus is now understood as a standard retroviral protein synthesis mechanism schema. So, we can put this schema alongside eukaryotic and prokaryotic schemas to understand something even more general about protein synthesis.

[10] See Glennan's work on ephemeral mechanisms for an account of one-off mechanisms, i.e., historical mechanisms that may occur only once (Glennan 2010).

In this way, Darden's suggestion is that general knowledge in biology is built by clustering related mechanism schemas in this way, where in our example we have two paradigm cases, closely related and covering many different kinds of cells—and we also understand that there are many non-paradigm cases, like retroviral protein synthesis, and probably many more quirky and particular cases to discover. The cluster cannot be reduced to a laws-description, nor can the local quirks be collapsed into an overall schema. Biochemistry students build knowledge by learning about the paradigm cases—and about the quirky cases. In spite of such variation, the collection of these schemas yields what general knowledge is actually available concerning protein synthesis. It is at least clear that understanding the cluster gives scientists something far more general than they gain from understanding the protein synthesis of one particular cell, or even one paradigm mechanism schema. So Darden's suggestion seems to offer an insightful beginning to an account of generality in mechanistic knowledge.

We turn now to use the work of Carl Craver (2007) on what he calls "mosaic unity," which develops both his joint work with Darden, and her early work—such as Darden and Maull (1977).[11] The traditional view of theory unification that we summarized above tended to assume that unified theories would be restricted to a single domain, and until relatively recently the dominant domain examined was physics. We explained Darden's suggestion above with reference to the life sciences as she studied them extensively, but restricted ourselves to the single domain of biochemistry. However, Craver notes that scientists increasingly collaborate across domains to explain phenomena, discovering what he calls "multifield mechanisms" (Craver 2007). He is interested in how work based in different scientific fields is *integrated*. Craver studies the integration of various sciences of the mind and brain, which offers excellent exemplars of what he has in mind. We are interested in his insights into how unity can be made that crosses scientific fields, rather than his more particular studies of neuroscience. Nevertheless, his choice of case is best explained in his own summary of an influential 1973 paper:

> [Bliss, Gardner-Medwin, and Lømo's] introduction is an extended argument for the relevance of LTP [Long-Term Potentiation] to learning and memory. Their argument, not coincidentally, appeals to results from multiple fields. They appeal to experimental psychologists' ablation studies ..., biochemists' assays of the molecular constituents of the hippocampus ..., physiologists' EEG recordings during memory tasks ..., psychiatrists' evaluations of patients with brain damage ..., electrophysiologists' theoretical considerations ..., and computer scientists' models ... . Results from these different fields constrain the possibilities for situating LTP within a multilevel mechanism. Craver (2007, p. 243, citations deleted).

[11] This part of Craver's influential book is not much discussed, although the fact that the idea of mosaic unity is the topic of the final chapter, and appears in the book's title, suggests that Craver considered it very important, at least in 2007. Note that while Craver creates his account by studying neuroscience—as Darden studied protein synthesis among other things—we are focused on his theoretical results, the account of mosaic unity.

We will assume, in accordance with minimal mechanism, that mechanisms are typically hierarchical in this kind of sense. The relevant interrelationships among our three infosec cases will become a clear example of it in Sections 3 and 4, showing a similar significant heterogeneity in relevant kinds of evidence. Craver argues that our understanding of memory has advanced, not by reducing psychological phenomena to neuroscientific law, but by understanding the relationships between many entities, activities, and their organization, drawing on all the disciplines listed above.

Craver suggests that we can understand this integration as multiple fields all exploring a space of all the mechanisms that could possibly explain the phenomenon of memory. Scientists never explore that whole vast space, however. Instead, what we know of different fields suggests "plausible" mechanisms. The discovery of new entities and activities, and forms of organization, can make some parts of that space implausible, or open up new areas as plausible: "A constraint is a finding that either shapes the boundaries of the space of plausible mechanisms or changes the probability distribution over that space..." Craver (2007, p. 247). Craver discusses many such constraints, but one of the easiest illustrations comes from spatial and temporal constraints (which Craver takes to operate both intra-level and inter-level). Time is important for memory, as it was crucial that what was possible at the cellular level—the strengthening of synapses to strengthen neural connections in LTP—could last long enough to be a plausible part of the mechanism for something that we knew, at the psychological level, could last a long time—human memory. In this way, knowledge gained from multiple fields is slowly, carefully, integrated to provide an understanding that crosses fields. Generality is a slow, and often painful, achievement.

This allies very naturally with Darden's work. Memory is not a unified phenomenon (Bechtel 2007), and we can think instead of a cluster of multifield mechanisms of memory—using Darden's terms alongside Craver's. Infosec similarly crosses levels, from the sociotechnical system mapped by the NCA, to the very technical areas of malware analysis, as we shall show, and can draw on knowledge from multiple disciplines, techniques and technologies, so that beginning from the combination of Craver's and Darden's work should be useful to apply to infosec.

Very recent work by Stuart Glennan suggests further insights that can be interwoven with those of Darden and Craver. His current view accords well with the view we have been building. He writes: "But scientific fields are not islands. For one thing, they are integrated by what Darden and Maull (1977) once called interfield theories. This integration does not come via a grand theoretical reduction, but rather by exploring localized relations of mechanism dependence, where entities or activities assumed in one field are located, filled in and explained in other fields ..." (Glennan 2017, p. 143).

We will accept this interweaving of views, and focus on drawing out some insights of Glennan's very recent work about kinds of mechanisms and ways of comparing them—his "taxonomy" of mechanisms. He uses the account of "minimal mechanism" we opened with to build it. The heart of his view is simple: each of the parts of the characterization of mechanism direct us, somewhat independently, to how mechanisms are more or less similar to each other. Three parts are obvious: phenomenon,

entities and activities, and organization.[12] Glennan adds a fourth, etiology, which is history, or how the mechanism comes about.[13] He writes first of entities and activities:

> The picture I have given of mechanism kinds gives us some understanding about the nature of disciplinarity and of the extent and limits of the unity of science. Scientific fields are largely defined by what I have called material similarities—similarities in what (material) kinds of phenomena they seek to explain, as well as the set of entities, activities and interactions that they take to be (potentially) responsible for these phenomena. ... Disciplines grow around the material and theoretical resources, technologies, and experimental techniques used to explore these phenomena and the mechanisms responsible for them (Glennan 2017, p. 143).

Entities and activities are often the most striking similarities and differences among mechanisms. Mechanisms that share common entities, like neurons, are obviously similar in sharing neurons, and fields form around the sharing of common technologies used to study those entities and their activities. This seems fairly obviously true, and indeed one way of thinking about the insight Glennan offers is that phenomena, organization and etiology are *also* very important to understanding and comparing mechanisms. Glennan offers us an account of these that we will draw on more later, and argues that these dimensions of comparison are to an important extent independent: for example, mechanisms for the same phenomenon might include quite different activities and entities, and mechanisms with similar entities and activities might have quite different forms of organization.

Let us here just briefly examine how Glennan's point can illuminate the idea of clusters of multifield mechanisms that we came to from examining Darden and Craver's work. We already have the idea that multifield mechanisms are painstakingly built up by scientists collaborating on understanding a particular phenomenon, and, given that mechanisms are not unified, there are likely to be multiple related mechanism schemas for a particular phenomenon. Glennan's work, then, helps us to see *four* different places to look for clustering among related mechanisms.

Of these four, relations between activities and entities in related mechanisms will be obvious. The shared technologies created to study them are likely to strike us. The nature of the phenomenon, as this is often the initial focus of work, is also likely to strike us. Forms of organization and etiology will be much less obvious. But if Glennan is right, and we think he is, we should expect them to be present, and particularly important to cross-field mechanisms.

We have, then, woven together threads of insight from Darden, Craver and Glennan into an initial picture of the building of general knowledge by discovering clusters

[12]The ideas of these as important dimension is already in the discussion of multifield mechanisms in Craver (2007), but Glennan develops this considerably.

[13]This is related to a skeletal account in Illari and Williamson (2012), and is summarized in Glennan and Illari (2017).

of related multifield mechanism schemas, that we should expect to vary along four dimensions: activities and entities, phenomena, organization, and etiology. We can see that in this array of ideas, the mechanisms literature offers us ways of thinking about what counts as such general knowledge as it is possible to get, and where to look for it within sciences which discover mechanisms. We will shortly apply these theoretical insights to infosec. We begin in the next section with a more detailed exploration of the challenges to building general knowledge in infosec, and how practitioners respond.

3 Building Mechanistic Knowledge in Information Security

Building mechanistic knowledge in information security faces many challenges. Some are similar to those addressed in the existing philosophical literature. We focus on three challenges that, while individually not unique to infosec, jointly produce distinctive difficulties in building general knowledge, making this a fascinating domain for philosophers interested in general knowledge to study. In Section 3.2, we explore three interrelated examples of active research problems in infosec that each demonstrate the triad of challenges we establish in Section 3.1. Each example overcomes the triad of challenges differently, and yet each can be illuminated by the picture of building mechanistic knowledge provided by the philosophical literature.

3.1 The Three Challenges for Information Security

Any problem domain has its own quirks that give practitioners difficulty. In experimental physics, building precise measurement devices to detect rare events is a challenge. In virology, pathogens evolve and change year-to-year, thwarting vaccines. In macroeconomics, one has to rely on natural, rather than controlled, experiments and cope with the fact that many of one's subjects are people, who may read and respond to research results and policy announcements. In this section, we will introduce three aspects of infosec that have heavily shaped research design and methodology in the practice. While none of these three aspects is unique to infosec, each exacerbates the other two. This subsection will provide a rough introduction to the infosec problem space and its problem-solving methods. As we shall see, this triad of challenges has forced infosec practitioners to refine their methods beyond what is expected in disciplines where each aspect or challenge arises alone. This triad does not cover all the challenges in infosec, but its combination provides an instructive set of examples.

The three challenges in infosec we focus on are: the immediate object of study, namely software, can change behavior during or between observations; active adversaries respond to, and try to confound, observations; and there is often justified secrecy among friendly parties. We will show that the combination of these aspects of infosec research pose notable methodological challenges. They are challenges for many aims of practitioners, but we devote time to explaining challenges not well known in the philosophical literature. In the next section we move on to illuminating how they pose a challenge specifically to the building of general knowledge,

by showing how some success has been achieved, using our three examples of mechanism schemas drawn from active research problems in infosec.

Changeable Software That software is changeable is a property of computer science generally, not just security.[14] Code is easily and often changed by human software developers, and running programs may change their own behavior during execution (Thompson 1984).

This challenge includes at least two closely related difficulties. First, the fact that humans frequently adapt code, and can design, reuse and redesign code to behave differently based on input parameters, will be relevant for our discussion of malware. Second, the fact that software environments are complex, such that code may behave differently in the presence of certain software packages, combinations of input values, or on certain hardware and not others, is more relevant to studies of reliability, vulnerability detection, and debugging.[15] The latter difficulty might be considered the dynamic nature of software, rather than its changeability per se. However, malware authors tend to leverage the ambiguity afforded by the dynamic nature of the software environment to their advantage. The salient aspect in which software is changeable is that argued by Hatleback and Spring (2014); not that the source code is editable, but that the behavior is both dynamic and designed to be responsive to the environment in arbitrary ways. Both the arbitrariness and having-been-designed make studying the dynamism of software a distinct challenge from dynamism in other fields such as chemistry and biology. For this reason, we use "changeability" to capture both difficulties simultaneously.

In computer science practice, one may want to verify that code written has certain properties or meets particular requirements. That software can change dynamically during the test or experiment is one major challenge in this endeavor. Different run-time results can be purposeful (for example, if today is payday, then pay employees), accidental (if I trip on the network cable while the program is talking to the bank, and disconnect it, it fails because the environment changed), or stochastic (for example, the program generates a random number to start from). One impact of these various changeabilities is that a lot of effort in software development is put towards testing whether a patch or update has re-broken an old fix of an old flaw. Such "regression testing" only approximates the ideal of testing each change against each past fix over each possible task the program might perform (Brooks Jr 1995). In practice the software changes too much to test all those possibilities exhaustively, both in that programmers make edits more quickly than are practical for tests and in that potential software run-time deviations based on task and environment are more numerous than

[14] For a review of formal semantics attempting to cope with changeability, see Winskel (1993, p. 297ff). Difficulties related to the changeability of software figure prominently in the historical development of the internet (Hafner 1998). To account for such changeability during mechanism discovery in security, Hatleback and Spring (2014) argue for heuristically dividing mechanisms into those that are engineered and those that are physical or natural.

[15] We thank an anonymous reviewer for suggesting to us this very useful way of making and describing this distinction.

are practical to test. Due to these facts, exhaustive testing of changes is not plausible and a different solution is needed to manage this challenge in infosec.

Deceptive Adversaries The first challenge becomes particularly pernicious when combined with the second challenge: active adversaries deliberately exploit the changeability of software, re-writing it to make it harder for defenders to detect and repel. To be an adversary, something or someone should be what is known in game theory as a bona fide player; that is it must "(1) make choices and (2) receive payoffs" (Rapoport 1966, p. 20). A *deceptive* adversary takes actions in response to the infosec researcher to try to change the researcher's conclusions.

Some methodological problems involving the target system altering during study are already known in philosophy of science. For example, experimental manipulations made on a target system can sometimes alter the causal structure of the system itself. Such experimental manipulations are known as "structure-altering interventions" in the literature on Woodward's interventionist theory. The problem is discussed beyond this literature, though. Mitchell (2009, p. 67ff) applies this to gene knockout experiments. These aim to find out what the knocked out gene normally does, but face the methodological challenge that genes are strongly interactive, and backup mechanisms exist for many essential cellular processes. So if you knock out one gene, another set of genes often activates to fulfill the task. This means practitioners need to find other ways to establish the role of the knocked out gene.[16] However, we will show that the combination of the problems of changeable software and deceptive adversaries goes far beyond that of structure-altering interventions.

Other domains also study targets that in a sense actively resist. For example, immunology in a sense faces adversaries in the pathogens they study. Pathogens do change in response to antibiotics and other treatments and environments. However, pathogens do not make choices in the way that adversaries do in infosec. They certainly cannot read the immunologist's research papers and figure out ways in which to subvert them, while adversaries can do this in infosec. This has a noticeable impact on building general knowledge in infosec, and leads us to our third challenge.

Well-Motivated Secrecy The third challenge of the triad pushes the overall problem further beyond challenges discussed with respect to other domains. Infosec practitioners must hide knowledge and successful strategies from adversaries, and so cannot freely share knowledge and successes. Not only does this need for secrecy lead to repeated work, but each infosec practitioner is not in sole control of what knowledge or successes are leaked to the adversaries, who then use that knowledge to instigate changes to their deception.

The three challenges that we investigate and clarify are averred by practitioners, including Kaspersky, an anti-virus and security-consulting firm, in a recent report.

[16] See extensive discussion by Steel (2008) and Cartwright, primarily with respect to social sciences. As Cartwright (2012) points out, within economics this problem is known as the "Lucas Critique," following Lucas (1976).

For example, on the challenges of secrecy among friends and active adversaries, consider:

> ...we remain bound by corporate realities, respect for the research methods of collaborators, and, most of all, legal constraints. As such, we may not always be able to provide full disclosure of indicators involved in certain findings. ...we feel these are not vital to convey the main thrust of our argument, which is that intermediate-to-advanced threat actors are aware of attribution methods and are already attempting to manipulate researchers to expend limited resources chasing ghost leads. Where gaps arise, let us relegate these accounts to camp fire re-tellings among friends (Bartholomew and Guerrero-Saade 2016, p. 3).

They also discuss the need for, yet difficulty in, constructing general knowledge:

> An often ignored facet of the [infosec knowledge] production cycle is the role of the analyst whose purpose is to coalesce various sources of information, arrive at various conclusions, and vet the overall logic of the finished product. Sadly, at this stage in the rise of the threat intelligence industry, deficient hiring practices overemphasize specialized technical knowledge and eschew generalist broad-thinking capabilities, often assuming technical candidates will bring these in tow. This is seldom the case... (Bartholomew and Guerrero-Saade 2016, p. 9).

This challenge of secrecy goes along with changeability and deception to create a particularly serious barrier to the building of general knowledge. Ultimately, if general knowledge is to help improve infosec practice, then it needs to be in a form that can be *shared*, as general knowledge is shared in many scientific fields. The need for some kind of shareability that meets these challenges becomes an integral part of the problem of building general knowledge in infosec.

The idea of sharing is worth pause for thought. One obvious way of sharing knowledge is to publish it in the standard academic, peer-reviewed venues. However, there is a spectrum of sharing between telling no one and publication, and multiple options are important to infosec. The spectrum ranges from fully-formed government classification networks with strict military-legal guidelines, to contractual non-disclosure agreements among corporations, to informal networks among peer individuals. Indeed, current attempts to reduce barriers imposed by secrecy predominantly involve painstaking networking among professionals in the field to build personal relationships that support sharing. There is not much research into this phenomenon, but Sundaramurthy et al. (2014) anthropologically documents a case study of the difficulty of gaining trust among computer security incident response staff.

Information sharing may also self-organize or be mandated. Two examples of self-organized or self-selected constituencies are the Anti-Phishing Working Group and Shadowserver. An example of mandated sharing is the US Presidential Decision Directive 63 which, in 1998, formed information sharing and analysis centers for each of the national critical infrastructure sectors. Game-theoretic analysis of information sharing suggests firms best voluntarily share information in the implausible scenario of highly competitive markets with firms both large and equally matched – and even then the results fall short of what would "maximize social welfare" (Gal-Or and Ghose 2005, p. 200). Modern operational data agrees that sharing is disjointed

and visibility partial.[17] Further, infosec contains what economists call a market for lemons: where a consumer cannot distinguish quality products from bad ones (lemons), though the vendor has the information to make the distinction (Anderson 2001).[18]

We will be interested in multiple possibilities for sharing beyond academic publication, but only the forms of sharing that are relatively wide. That is, we are interested in what can be shared beyond painstakingly built trusted private networks.

3.2 Three Examples of Information Security Mechanisms

In this subsection, we will illustrate how practitioners approach these common challenges through three examples of active research problems in infosec. The three examples we discuss here serve to justify and deepen the various assertions we have made above about the nature of infosec practice, particularly indicating the range of applications with which infosec contends. We explore infosec through (1) research to track and reduce the harm caused by the myriad attacks that steal money; (2) the intrusion kill chain model of an individual attack, which models the entities, activities, and their organization by which an adversary initiates, executes, and makes use of an attack; and (3) tools for reverse engineering a single piece of malicious software (malware), which is a particularly important entity in many individual attacks. These three examples form what is in some sense a hierarchy, (although, as we have said, what sort of hierarchy and what, if anything, "levels" are will not concern us). In reverse order, malware is almost always used in an attack, but it is only one part of the mechanism modeled by the intrusion kill chain. Likewise, various attacks are used in the mechanism of electronic crime (e-crime), but they are in turn only a part of e-crime considered more broadly, from the perspective of national agencies. In this way, the three examples are clearly hierarchically related. Taken together, they also demonstrate the scope of infosec, from social and economic systems through the technical minutiae of how malicious software takes control of a computer.

It will also become clear that infosec practice does still manage to achieve considerable success in spite of the three challenges, and we will use this subsection to show how thinking of the building of general knowledge in the field as the building of mechanism schemas—*shareable* ones—is a reasonable and useful way of conceptualizing the achievement. This will set us up to indicate, in Section 4, how fruitful this conceptualization could be for practitioners.

Botnets—the NCA's Banking Trojan Model Our first example comes from the UK's National Crime Agency (NCA)[19] and their description of how networks of compromised computers (botnets) are created, monetized, and the money laundered. The

[17] Dozens of lists of malicious computer locations are broadly disjoint, even across wide spans of time (Metcalf and Spring 2015; Kührer et al. 2014). Limited sharing also appears to perform worse than public sharing on website compromise recidivism (Moore and Clayton 2011).

[18] For further reading see the long-running Workshop on the Economics of Information Security (WEIS) or the Workshop on Information Sharing and Collaborative Security (WISCS).

[19] Thanks to Stewart Garrick for this example and permission to use his diagram.

NCA may seem an unlikely source for academic lessons on mechanism discovery. However, infosec concerns are endemic to all sectors of society, and much research activity happens outside academia. Figure 1 displays the "banking trojan business model" for internet criminals who steal money from consumer banking accounts, as described by Addis and Garrick (2014).

This criminal business mechanism is complex. It starts in the top-center of the diagram with the controlling coders, the software developers who create the software necessary to manage a diverse infrastructure of loosely coordinated compromised machines. This infrastructure is unreliable to the criminals, because the machines' owners may turn them off, move them, change their network addresses, or notice and fix the malware infection. The NCA is not concerned with individual computers in the network, but with the network itself: that it is unreliable to the criminals changes how they behave and what they build, and so what the NCA should look for. In particular, the criminals outsource various aspects of their business, like any other savvy project manager. Grier et al. (2012) and Sood and Enbody (2013) survey these exploitation-as-a-service and crimeware-as-a-service businesses. The various entities to which services are outsourced are listed in the diagram as "traffic sellers," "malware servers," and "proxy layers," for example. The NCA's mechanism discovery has decomposed the criminal's task into these entities. A security expert would also understand the activity localized to each entity. For example, traffic sellers use

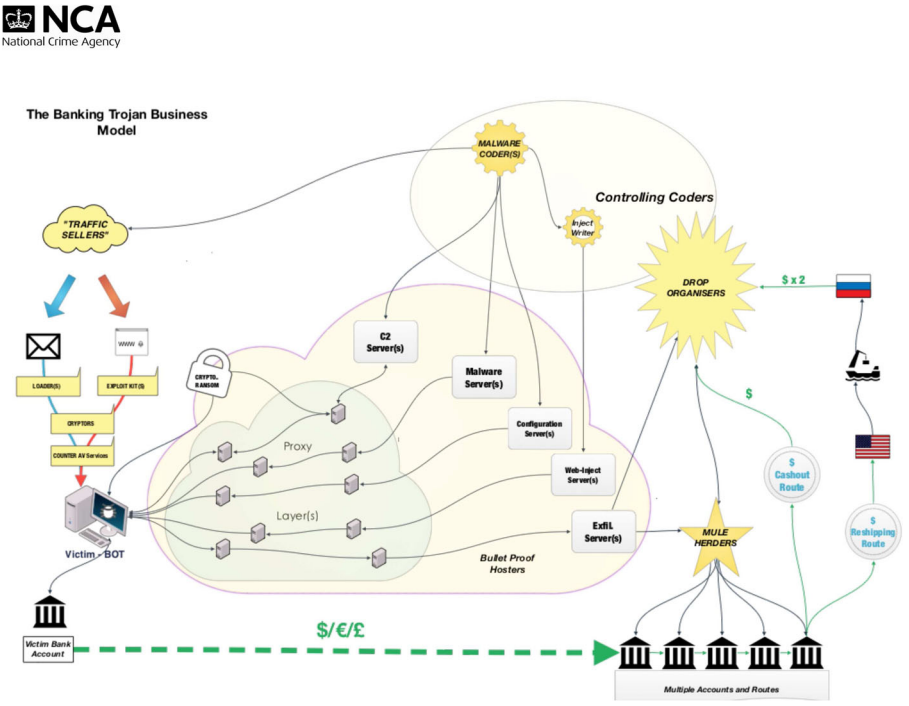

Fig. 1 Botnet theft and money laundering mechanism as described by the NCA in Addis and Garrick (2014). Reprinted with permission

various tricks such as compromising popular websites and sending unsolicited bulk email (spam) to direct potential victims to the controlling coders' malware. These two activities produce malicious emails and websites, entities represented underneath the "traffic sellers" entity. And so on with the other elements of the mechanism, leading counterclockwise eventually to the criminals receiving money.

In this way, the NCA's mechanistic model conveys a great deal of information. But on closer inspection, we can see that the NCA are also safeguarding against some of the challenges of infosec; specifically, the challenges of the changeability of software and the presence of active adversaries who will respond to published observations. The goal of understanding the mechanism of the criminals' business is to interrupt that business, and this goal could be impeded by publishing too much. Given this, the mechanism description focuses on essential functionalities. Although software is changeable, the internet's basic rules of operation do not change quickly. The Internet Engineering Task Force and Internet Architecture Board oversee change proposals, and key services may be updated only once a decade. The criminals must accomplish certain tasks within this framework, because all their potential victims are on the internet. Therefore it is relatively safe to publish to the criminals that the NCA knows traffic is delivered via email, websites, and proxy layers. There may be myriad ways to create software that performs these activities, but each activity itself cannot be easily abandoned if the criminals still want to accomplish their goal.

When legal authorities plan to intervene on a criminal mechanism of this kind, they must also respect the challenges of infosec. In examining the planning of successful interventions, one starts to feel the pressure of justified secrecy among friendly parties. As one example, we could imagine that Internet Service Providers (ISPs) detect indicators of compromise among their customers and notify the banks to freeze the account if one of their mutual customer's computers is compromised, thus limiting theft. However, privacy laws generally prevent ISPs from providing information about their customers' traffic to anyone, including banks. The ISP may not even be allowed to know the customer's bank. And where the victim's traffic is encrypted the ISP may not be able to detect when a customer is victimized at all. Encryption is mathematical secrecy between two parties. Encryption is a highly recommended protection against, among other things, criminals stealing your banking credentials during online banking. But encryption works just as well for the criminal to hide their attacks. If encryption is actually to provide privacy, intermediate parties like ISPs must not be able to distinguish between any two encrypted items, even if one is encrypted banking and the other encrypted attacks. So privacy, both legal and technical (provided by encryption), limit the possible infosec interventions.

For a crime agency, the end goal is usually to arrest the criminals. This seemingly straightforward goal is further hampered by a combination of the internet's global reach and international politics, which creates justified secrecy in another form. We all access (essentially) the same internet, whether it is from London, Moscow, or Tierra del Fuego. Arrest warrants do not have such an immediate global reach. Although mutual legal assistance treaties often succeed eventually, and there have been successful arrests, national legal processes do not allow broad sharing of suspects of investigations with just anyone. Further, the internet is dominated by pseudonyms, and arrest warrants for "xxCrimeBossxx" or "192.168.6.1" are not

terribly effective. Although private companies may know the identities of their customers behind these pseudonyms, for legal or contractual reasons private companies may not be able to share these with law enforcement, especially foreign law enforcement. This all means that mechanisms of intervention tend to focus effort on protect and prevent activities.

We can summarize how the NCA navigates the triad of challenges of changeable behavior of software, reactive adversaries, and justified secrecy. First, they diagram the relatively unchangeable constraints criminals *have* to operate within, and second, they publicize only constraints already known to criminals, and not alterable by them. Of course, this does not eliminate attempted deceit by criminals, and as we have seen, many issues of secrecy even among those attempting to preserve security still remain.

We will shortly turn to our second example, computer network attacks, but we will first just note the relations among our examples. As we have said, we take our three cases to form a mechanistic hierarchy in the sense described by Craver (2007). At the heart of the internet banking crimes described above are the computer network attacks we describe next. These are attacks which convert a healthy computer controlled by its owner to an infected victim controlled by an adversary. Attacks occupy the left-hand side of Fig. 1, from the traffic sellers through taking control of the victim computer, known as the "bot," and ending at the objective of access to the victim bank account. However, Fig. 1 does not detail how the attacks happen, what methods the criminals use, or who is targeted. In part, this is because the attacks used are diverse, and changeable, and so are hard to model. More importantly, for the level of the explanation of the criminal business model the details of how the attacks occur are not important. However, from a different perspective, of computer owners who would like to protect themselves, the details of how each attack happens are crucial to detecting and preventing attacks.

Descending a level further we come to our third example, malware analysis. Note that for our cases, malware is not placed at a lower level merely because it explains physically smaller items, or merely because a part is spatially contained within a whole (two popular views). Instead we follow Craver (2007, ch. 4-5) in holding that levels of explanation are relative to levels of mechanisms for the phenomenon of interest where the elements are indeed parts of wholes, but they are also mutually manipulable in the sense that changes at the level of the part will at least sometimes make detectable changes at the level of the whole, and changes at the level of the whole will at least sometimes make changes detectable at the level of the part. So these examples form a hierarchy because one of the components of the mechanism describing the criminal business model shared by the NCA is computer network attacks. And one of the elements of computer network attacks in turn is malware. This is not strictly a part-whole relationship; attacks can happen outside of crime, for example during nation-state espionage. And to explain even one malware sample used in an attack, one must explain not only its attack role but also its historical relation to other malware as well as how it hides itself. Yet in this way, a mechanistic explanation of attack adds to the higher-level explanation of the criminal business model, and vice versa, and so on with malware related to these two examples.

Computer Network Attacks—Lockheed Martin's Intrusion Kill Chain With that loose approach to mechanistic hierarchy in place, we can move down a mechanistic level. Understanding models of attack is our second example of an active research problem in infosec. One popular model of the steps any adversary must take in a successful attack is the intrusion kill chain (Hutchins et al. 2011). Spring and Hatleback (2017) argue that the kill chain can be considered a mechanistic explanation of an attack. The kill chain model decomposes an attack into seven steps, which in this case are most naturally understood as activities. For an individual attack, where an attack is defined with a quite small scope of targeting exactly one computer, these steps occur in a linear sequence. The seven steps are: (1) *reconnaissance*, gathering necessary details on a target; (2) *weaponization*, creating a malicious file suitable for the target; (3) *delivery*, sending the weaponized file, usually via email, web traffic, or USB drive; (4) *exploitation*, initial attempt to take over a computer once the file is delivered and opened; (5) *installation*, adding additional software to maintain a robust covert presence; (6) *command and control*, any communication between the installed malicious software and the human adversary for reporting, updates, or direction; and (7) *actions on objectives*, where adversaries finally move to complete their material goals. Adversary goals may include stealing files, corrupting essential data, or starting back at reconnaissance (1) to conduct new attacks which are only viable from a computer which the defender still trusts (Hutchins et al. 2011, p. 4-5). This describes a single attack, but note that an adversary almost always coordinates multiple attacks to achieve an effective campaign. Figure 2 captures the organization of the seven steps of the kill chain and the relevant entities.

The kill chain model avoids the challenge of the changeability of software and adversary responsiveness with a strategy similar in some respects to the criminal

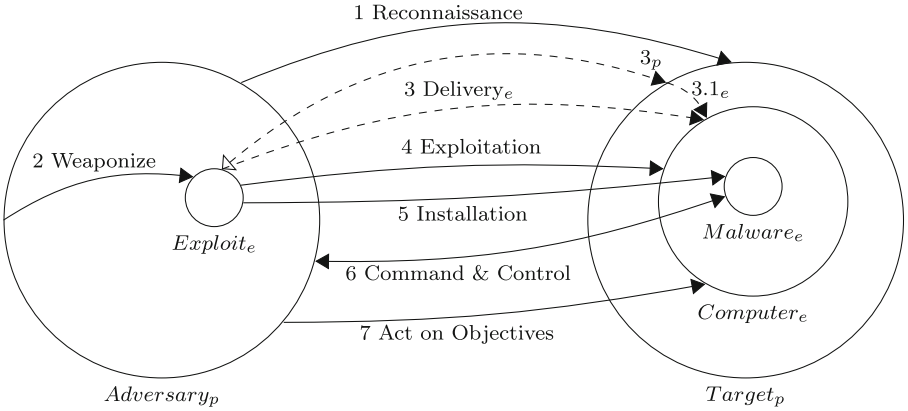

Fig. 2 Kill chain diagram, following Spring and Hatleback (2017). Circles are entities, arrows are activities, where delivery (3) can take two paths, one targeting the computer, and another targeting the human (i.e., social engineering such as phishing). The large entities are the (human) adversary and target; the medium-sized entity is the computer owned by the target; the small circles are a weaponized exploit and some malicious code, both written by the adversary. The subscripts are *e* for engineered and *p* for physical, following the distinction suggested by Hatleback and Spring (2014). If all salient elements are changeable by a designer, the element is considered engineered. Software in the primary engineered element. Aspects such as human cognitive biases that make us susceptible to trickery are considered physical

business model. The kill chain model contains somewhat abstractly specified activities that are necessary steps in a successful attack. There is extraordinary variability in the entities that perform these activities, where they are performed from, and the exact details of how to accomplish the activities, but the activities themselves are fairly stable. For an individual attack, the organization is also fairly simple: activities occur in linear order. That is slightly more complex for the usual case, which involves multiple interrelated attacks, which often run simultaneously. Nevertheless, the kill chain mechanism is a rare statement of a stable organization of stable activities even in the face of considerable variability and even changeability in entities. Adversaries still attempt to hide their activities and status along the chain, for example by conducting many simultaneous attacks at asynchronous stages of progress. For example, adversaries are known to confound attack response by quietly hiding an important attack within a noisy but merely annoying attack. Although they can cover their tracks in such ways, the kill chain remains a stable organization of activities per attack.

The triad of challenges in infosec all impact modeling at the level of the kill chain. The kill chain model was published by Lockeed Martin, one of the largest US defense contractors, an organization attacked by all manner of adversaries, foreign-government and otherwise. The Lockheed researchers who created it based the kill chain model on their own experience investigating and responding to attacks over eight years.

One common criticism of the kill chain model is that it is too abstract. This criticism directly relates to how few commonalities there are among these eight years of attacks. The changeability of software forces the level of analysis up to where the activities and their organization are both stable. But this level of analysis is too high, too abstract, for much day-to-day infosec because reasoning with the kill chain is not automatable. The kill chain research resists the active response of adversaries by selecting elements which the adversary cannot change and remain effective, but this resistance comes at a cost.

Defenders benefit from using the kill-chain model of the mechanism of attack because it is a focus for orienting a defensive posture and incident response based on what steps of the kill chain the adversary has accomplished. Although the kill chain alone is too abstract to actually be capable of detecting malicious software, it directs how the defender responds after detecting malicious software. Such models speed communication among defenders, who are almost always organized into teams, specifically, a computer security incident response team (CSIRT). Alberts et al. (2004) describe a detailed set of processes for incident management by CSIRTs. Several of these include establishing or coordinating clear communication, and educating constituents. Established models play an important role.[20]

[20]There is debate about whether functional explanations (which in a sense provide only a breakdown of tasks in a way rather like we describe here) are distinct from mechanistic explanations (Craver and Tabery 2017). An important distinction in that debate is whether the breakdown of tasks is considered complete, or as a stage on the way to something more distinctively mechanistic. This is a tricky case, because the kill chain model, as publishable and shareable, stops at describing activities. Nevertheless, we believe that, as it is generally used as a device for helping to identify the entities used in a particular attack, it is best thought of as a model of a mechanism.

Publishing the kill chain model helps to diminish secrecy among friendly parties by providing a common language of discourse and instructing defenders what to look for. There is a small risk that unskilled adversaries could use the model as a how-to guide. However, this risk of improving the skills of the least effective adversaries is weighed against the potential collateral damage of secrecy. Broad use by allies is also weighed against consulting profits of maintaining a proprietary model. Lockheed Martin is not a university; rather, they are publishing the kill chain as industry experts, to try to help their allies improve their defenses.

Malware Analysis Security cannot be achieved on the basis of the kill chain alone, though, because automation is a key aspect of effective infosec. The actual detection or prevention procedures are handled by computers. Even a small computer network handles billions of decisions every minute; no human can be directly involved at such a speed. Thus, the end goal of infosec work is often a pattern or indicator of malicious activity which is highly reliable and simple enough to be quickly checked by a computer. Failing to provide direct progress towards this goal is one criticism of the kill chain. Automation involves moving down a level of mechanism, although this sacrifices some of the stability achieved by the kill chain model. Malware analysis, our third example of an active research problem, is one method of generating such patterns or indicators.

In general, to find a pattern or indicator that will work as an adequate detection procedure requires a person. There is no hard and fast reason for requiring a person, but one important factor is the fact that the adversaries are people. Computers can sometimes develop adequate detection indicators; this is a common application of machine learning. Our example, malware analysis, is driven by a person. Malware analysis relates to our other two examples in that both have malware as a component entity, and the malware analyst is attempting to discover the lower-level mechanisms of how the malware functions. Research on these lower-level mechanisms contends directly with the challenges injected by the changeability of software and the adversaries' ability to respond to and interfere with research results.

Malware analysis is one example of many possible processes of building a method for understanding and detecting a specific attack or campaign of attacks. Roughly, a representative process is that a human malware analyst receives an unknown computer program that has been deemed suspicious or likely malicious.[21] The malware analyst then behaves much like a scientist. She will put the unknown sample in a specially designed, controlled environment. She then attempts to determine some relevant properties of the malware, and trigger the malware within the safe environment to exhibit characteristic malicious behavior or divulge information about its author or controller. Yakdan et al. (2016), Lin et al. (2015), and Lawrence Livermore National Laboratory (2016) describe some of the available technical tools for these tasks.

[21] The determination of what programs are suspicious is an independent topic. At a high level, a program is suspicious if it has properties similar to other malicious programs (attached to similar emails, for example) or if incident response staff unexpectedly find it on a misbehaving machine.

It is key to understand one fact about computers that makes this task difficult. Recall that we explained in the introduction that in practice and in principle, one cannot know *a priori* what a computer program will do. This is true even if you write it yourself, and the problem is far worse if an adversary wrote the program to be covert or confusing. The in-principle side of this argument is provided by a suite of formal results, including Turing's famous result that we described above (Turing 1936). More practically, when a malware analyst receives a program for analysis, it is just a blob of uninterpreted ones and zeroes. You may as well be asked to determine, *a priori*, whether a sentence in a foreign language mentions a cat or not. Even if you can determine the usual word for cat, the speaker may use any number of metaphors, synonyms, cultural references, or proper names to refer to a cat without using the word (such as "Felix" or "witch's familiar"). Computer languages can be similarly evasive. Colloquialisms or oblique references can conceal the subject of conversation—namely, malicious actions—from the software analyst. Because a computer tracks references with a precision impossible for a human, computer languages can also be arbitrarily long-winded and round-about while performing these oblique concealments.

Malware analysis comprises multiple specialized sub-tasks for defeating the deception methods that adversaries employ. A common deception is to hide, or obscure, the true purpose of the malicious software. There are many techniques adversaries use for hiding or obscuring, collectively called 'obfuscation'. Obfuscation takes advantage of the changeability of software to enable a broad class of activities, such as those described in the kill chain and the criminal business models, while attempting to avoid notice. The main task of a malware analyst amounts to seeing through these obfuscation techniques to discover the malicious mechanism that is the intended purpose of the malware. O'Meara et al. (2016) describe two case studies of malware development patterns over several years. The overarching pattern is that defenders defeat the common obfuscation technique at the time and publish a preventative measure, and then the criminals change their software to re-hide their larger-level activity of stealing money so that their thefts are successful again.

An illustrative example of an aspect of the cat-and-mouse game is to play with time. The malware analyst has thousands or millions of suspicious files to analyze. The malware authors know this. The authors also know that their actual targets likely will not know they have been targeted, and tend to be on their computers for a while after the initial infection. So one of the early tactics the malware authors implemented was to make their software sleep, or incubate, for two minutes before doing anything. This defeated malware analysts who opened a file and expected malicious behavior immediately. Some analysts figured this out and realized they could wait. Then the malware authors increased the sleep period to an hour, far more than any analyst has time to wait, even in mass-automation of analysis. However, the malware analyst totally controls the environment, so they can move the computer environment's clock forward 12 hours and trick the malware. The malware authors realized this and started using arithmetic instead, basically telling their malware to count to a trillion by ones before acting. While counting is notionally benign, malware analysts soon realized that there are not any benign programs that start and just count for a while, so this in itself becomes suspicious. And so on, strategy and counter-strategy.

Under these difficult circumstances, provenance, or the historical sources and similarities among malware files, is often the most useful guide. Groups of attackers tend to reuse their past work rather than start from scratch. The similarity of a malware sample to past samples tends to be important for understanding it. The history is the most stable part of the target mechanism.

In these examples, infosec practitioners find ways to overcome the joint challenges of the changeability of software, justified secrecy, and active adversaries. As the malware analysis example exemplifies, the combination of these challenges is particularly pernicious. If the adversaries could not make malware changeable in a way reactive to practitioners' analysis attempts, understanding adversaries' activities would be less daunting. If practitioners could share detailed results widely without tipping off their adversaries, this daunting burden could be shared and made easier. Alas, this is not the case.

In all three of our examples, infosec practitioners build and use knowledge of stable activities, stable organization, and the properties of entities—recognizably mechanism discovery strategies. These mechanism discovery strategies overcome the three challenges and build general knowledge, though practitioners rarely use these words. Our three examples each focus on a different aspect of what could be thought of as a multifield mechanism which collects relatively stable and shareable knowledge. Within the banking trojan example of money laundering, the organization of the parts of the mechanism is the focus; details of the entities and activities are secondary and remain abstractly specified. The intrusion kill chain provides a schema of activities that attacks contain, roughly organized in linear sequence, largely independent of the highly changeable entities involved. When studying malware, analysts are often interested in provenance, which equates to etiology or the historical sources of a malware file. Groups of attackers tend to reuse their past work rather than start over; similarity to past malware samples provides important understanding. While each of these examples builds its evidence through different strategies, they also mutually reinforce each other as part of a multifield mechanistic explanation. The following section expands on the philosophical theme of building general knowledge in infosec.

4 Building General Knowledge in Information Security

This section turns to demonstrating how the philosophical threads drawn from the mechanisms literature in Section 2 can illuminate the field of information security. We will view the various strategies used by infosec practitioners as mechanism discovery strategies used in the face of the triad of challenges. This allows us to see some coherent purpose behind what at first sight are wildly different actions by practitioners.

The examples elaborated in Section 3 demonstrate the depth of the triad of challenges for building general shareable knowledge in infosec. There is no stable system into which one can make surgical interventions in the way of Woodward. The challenge is far more difficult. Even further, what infosec practitioners are facing spans from the technical, in malware, to the social in the NCA diagram and international legal systems. Flechais et al. (2005) and Anderson and Moore (2006) claim that

only by considering the sociotechnical system as indivisible can we make adequate security evaluations. This, along with the fact that many parts of that sociotechnical system are highly responsive, means that infosec is in a constantly evolving arms race between defenders and adversaries.

The examples in Section 3.2 also contain myriad successful responses to that challenge, and began to draw attention to how success was achieved and then publicized. We can now extract what successful responses share. In particular, we will show that the strategy in all three cases was to find what is relatively fixed in the midst of a changeable social and technological system, and, of that, what is publicized is what need not remain secret. What remains fixed varies in the different cases, but we will show that they can be seen as elements of mechanisms, in the sense discussed by Glennan. Further, just as in other scientific and technical fields, mechanisms in infosec do not stand isolated and alone. Mechanisms both cluster within fields and are interrelated across fields, such that we can see our three examples as parts of a multifield mechanism, as recognized by Darden and by Craver.

We can productively see synthesizing general knowledge as linking up mechanisms along these complex and highly variable lines. Indeed thinking of this as modeling mechanisms illuminates the coherence of that search for shareable general knowledge. Finally, infosec both shares enough features with other fields which perform mechanism discovery that we can use the mechanisms literature; while it has enough peculiarities to help develop that philosophical literature in interesting ways.

Four Elements of Mechanisms Yield Four Dimensions of Variation *(Glennan)* First, we will show how paying attention to the four dimensions of similarity of mechanisms that Glennan draws attention to lets us see how each of these very different examples can be seen as cases where practitioners search for some stability in the mechanism. Glennan's four dimensions for searching for similarity among mechanisms are the entities and activities involved, the organization of the components, the phenomenon for which the mechanisms are responsible, and the etiology or history leading to the mechanisms (Glennan 2017).

Our three infosec examples each focus on a different aspect of mechanisms. Tracing banking trojan delivery follows the organization of the parts; modeling the adversary's kill chain follows the common activities involved across incidents, and also shows a reasonably stable organization; and finally malware analysis focuses on the history (which Glennan calls etiology) of how the file and its mechanism came to be. In all three cases, the practitioners use these as foci to work on other things they need to know. Note that none of the three cases we consider focuses on entities. This might be partly a feature of the cases we consider, of course, but we suspect that, at the least, it reflects the changeability of the entities. To deal with this, practitioners look elsewhere to coordinate a response, and find out how to stop a particular piece of malware, attack in progress, or criminal organization.

This helps to explain why the surface actions of practitioners can be so very different, although there is a sense in which they are all working on the same very general problem. Notice also that it helps to explain why each example coalesces under a recognizable discipline within infosec: internet architecture (banking trojan delivery),

incident response (kill chain), and reverse engineering (malware analysis). Practitioners acquire considerable expertise within their fields. Nevertheless, each can be seen as focusing on an aspect of the highly changeable and reactive mechanism they are facing. And each makes the sensible choice of focusing on what remains most fixed, and of sharing information about what does not need to remain secret—because it is already known by, or cannot easily be changed by, the criminals.

These Mechanisms are Multifield *(Craver)* Another important way of seeing the coherence here is to understand the models of mechanisms infosec deals with as hierarchically related, specifically multifield in Craver's sense. Craver (2007, ch. 7) argues that in the multifield research program of neuroscience, explanation of memory is best understood as a mosaic of interlocking evidence of a mechanism spanning multiple levels. The different fields locally depend upon each others' evidence. Each field provides support to the other; one is not reduced to the other (Kaiser 2011). Seeing the three examples we have used as combining into a multifield mechanism in this way is also useful for understanding how infosec succeeds.

Our examples of attack and crime modeling provide one example of this interlocking support. The kill chain model describes activities carried out over these distribution channels to which the adversary is constrained. Kill chain terms such as *delivery* and *exploit* describe steps in the banking trojan ecosystem at a finer level of detail. On the other hand, the banking trojan model expounds on the kill chain's final activity, *action on objectives*, to fill in what the objectives are (steal banking credentials) and explains how criminals use short-term objectives as stepping stones to accomplish their overarching mission. In this way each field supports the other; neither has primacy.

So the interrelationship of these three examples occurs in several ways. The criminal business model is a higher-level mechanism because the kill chain mechanism is contained within the criminal process. In another sense, the kill chain represents a mechanism schema which is partially instantiated by the criminal business model. The crime model instantiates the kill chain because it restricts certain kill chain activities, such as delivery, to the more specific methods of malicious websites and phishing emails. The kill chain activity of command and control is also instantiated. The NCA criminal business model is specific to a particular botnet; in this specificity it makes sense as an instantiation of the purposefully-general kill chain model.

None of these relationships are strictly part-whole, nor is malware solely deployed to steal money. Nevertheless, for understanding this particular criminal activity—and stopping it—we have to understand this multifield mechanism, forming a loose hierarchy where what we know about each level constrains how the whole can operate in the way that Craver describes.

Clusters of Related Mechanisms, not Unitary Mechanisms *(Darden)* While we chose to examine three mechanisms that were interrelated in a way that illuminates the multifield and hierarchical nature of infosec practice, it would be a mistake to think these are the only mechanisms in their domain. Instead, there is a great deal of clustering of related mechanisms, in accord with Darden's work. The models of the NCA mechanism, the kill chain mechanism, and malware reverse engineering mechanism are

each quite abstracted, capable of being filled in in different ways. So each is better thought of as an exemplar, allowing the practitioner to understand a cluster of related mechanisms.

Multifield mechanism models are a focal point for collecting and anchoring general knowledge. They do not describe one unified mechanism, but a cluster, or exemplar of related clusters. Alongside the other two mechanisms, the NCA's model of the mechanism of computer network attacks form part of a cluster that illuminates the phenomenon of a criminal campaign to steal money using the internet. We elide the technical details, but the law enforcement action included deceiving the criminals' communication, public awareness to help prevent and detect attacks, and seizing key physical computers in 11 countries simultaneously (Addis and Garrick 2014). The successful execution of these interventions indicates practitioners developed sharable, general knowledge; we conceptualize these three examples as a loose hierarchy, and also within the cluster of multifield mechanisms forming this general knowledge.

For example, the kill chain model provides a schema about which to cluster other mechanisms of other specific botnets. Both the original kill-chain work, and further work building on it, use the kill chain as a schema about which to cluster malicious activity for attribution of similar attacks to similar actors (Caltagirone et al. 2013). Infosec practitioners doing attribution of attacks cluster on targets, techniques and procedures, and malicious infrastructure. There is clear resonance here with the clustering features described by Darden, along the dimensions explored by Glennan.

Glennan (2017) can be used to illuminate how clustering works. Clustering mechanisms requires a feature on which to cluster. Darden and Craver make the case for hierarchy, but do not tell us much about *what* about a mechanism is similar to another mechanism that permits building general understanding. Hierarchy is not enough to explain on what dimensions mechanisms are similar. Glennan's dimensions provide features on which to cluster, or features to guide or assess multiple fields investigating similar mechanisms.

So we can see that there are interesting relationships within the picture of infosec we have drawn. These include the four elements of mechanism, clustering of mechanisms within a field, and hierarchical relationships across fields. These are all differences we can see in infosec. However, the distinctions are not sharp. Perhaps they never are, but with a domain as fluid, flexible, and reactive as the changeable technologies and social systems of infosec, we should not expect to find sharp distinctions and rigid boundaries. Whether a particular difference counts as variation in, for example, an activity, a different activity, or a schema instantiation, may often be indeterminate. This does not mean we cannot usually say something useful about relationships between neighboring activities, or between an activity and organization, especially when we can specify a context and a purpose.

4.1 Constraining: on Improving Coordination in Infosec

The resemblance of mechanism discovery in infosec to that in other disciplines is very useful. We will now try to indicate how this work might improve coordination in infosec by seeing that the disciplines we have studied collaborate to generate

understanding by adding *constraints* on the overall mechanism of online crime. As we have noted, Craver describes this happening in neuroscience (Darden and Craver 2002; Darden 2006; Craver 2007). So we are thinking broadly of our development of Craver's view that the discovery of new entities, activities, forms of organization, and etiologies can open up—or close down—space in the overall space of plausible mechanisms. This is how we are suggesting viewing how discoveries in one of our case studies can impact on others. Seeing these relations across very different cases can show how knowledge in infosec is available that is more general even than the knowledge we have indicated can be built of each case on its own.

Let us begin with malware analysis, which relies on constraints from our other two examples for building general knowledge. For those families of malware that steal banking credentials, there is a remarkable coupling between defenses by the financial services industry and malware capabilities added to circumvent those defenses (O'Meara et al. 2016). Financial institutions add social defenses that interrupt the business model. For example, sending PINs to customers' phones; malware authors quickly learn how to infect the phones and forward the PINs. Malware is also often categorized based on what stage of the kill chain it is involved in: initial exploit, permanent presence, command-and-control infrastructure, or achieving objectives. The very name banking trojan uses such a categorization: trojan malware is for a quiet, permanent presence, and "banking" indicates what sort of objective it is used to achieve. Other malware might be categorized based on what vulnerability it exploits, for example. So, if you know what must remain stable on one of the other levels of the hierarchy, that constrains where you should look in your efforts to combat malware. Knowledge at one level is a guide to help to build knowledge at another level.

Malware analysis likewise supports and constrains kill chain analysis. A particular malware file can only run on a specific system, say a Windows PC or a web server. By indicating what the malware could possibly target, the malware analyst constrains what the potential *delivery* and *action on objectives* activities are. The kill chain model constrains where a practitioner might look to find malware; if a computer has been the recent target of reconnaissance, it is more likely malware has been delivered to try to exploit the computer. Glennan's work can help us illuminate how constraints work, by finding the feature which is constrained, and in particular where that constraint will travel to related mechanisms. For example, if we know what vulnerability (an entity) that some malware exploits, we can constrain our search for infected computers to those running vulnerable versions of that software.

The fixedness practitioners seek in the midst of the triad of challenges tends to occur at the boundary of the phenomenon of interest and another field or system that provides constraints. The sociological mechanisms[22] of how humans interact with

[22] Social systems fall under what Hatleback and Spring (2014) call physical mechanisms, in that they are not purposefully designed. Social systems have been discussed as mechanisms for some time (Elster 1989). Social and legal systems seem a liminal case between the changeability of engineered mechanisms, such as software, and the more fixed nature of physical mechanisms in fields like chemistry. However, the salient feature is that at least some of the relevant social systems are much less changeable than malware. The local dependence of the malware on the social provides a dependable constraint on the changeability of the malware.

the technology available through the internet are email and web browsing; thus these are the constraints on the distribution channels of banking trojans. The challenge in the case of the banking trojan business model is to determine how the various mechanisms of theft are organized. It turns out the organization of the mechanism tends to be similar even for different criminal organizations, if the infosec practitioners look to where the various criminals face similar constraints. The criminals must distribute via the channels their victims are used to. The kill chain provides constraints on the activities, delivery before exploitation, for example. Internet architecture provides constraints on the entities for delivery, web and email. Criminological work such as the NCA model constrains the organization and can localize the elements of the mechanism: both web and email are used simultaneously in this case, and the task is outsourced to specialized individuals by the criminals. In this way, understanding how the three mechanism schemas (or clusters of schemas) we have described relate clearly yields much more general knowledge than understanding one alone.

At the other end of the multifield mechanism, we can also see constraints operating in malware analysis. Here the changeability of entities is very marked, and a technological arms race exists between criminals and infosec practitioners. Nevertheless there are important constraints. Hierarchy matters: malware is an entity within the kill chain; the activities of the kill chain are activities that the criminals undertake in their business. However, this overview does not adequately capture the nuanced similarities between these mechanisms on other dimensions. The criminal business model is specific to a criminal network named for the malware used to make the thefts: GameOver Zeus. The malware analysis that provides this name, specifically this family name of Zeus, is based on a etiological cluster of many malware samples which all have a common progenitor malware (O'Meara et al. 2016). Attackers automate and outsource where they can, but they are ultimately people, and can only build on what they have done before. And so the etiology of the attack, what is known of that source, is a useful guide to defenders.

Our three examples are, in some sense, grouped around similar entities or activities investigated through different lenses. We have tried to show that constraints that we learn about in one activity or entity here can impose constraints on the whole. We have also indicated that not only entities and activities, but also organization, phenomenon and etiology are important.

If this is right, then what we understand about constraints hinges on being able to home in on one of the dimensions of similarity identified by Glennan (2017). When we see that similarities among mechanisms extend to four dimensions of variation, we can see how the constraints work. There is no simple link between types of similarity among mechanisms and relationship between mechanisms, either within the same level or at different levels. Nor is there an easy link between aspects of interfield mechanistic explanation, i.e., mosaic unity, and similarity among mechanisms. However, for two mechanisms to be related, or two fields to interrelate, they must be related by something. These four dimensions of similarity provide a plausible starting point.

Ultimately, in the face of these challenges, infosec practitioners have achieved a great deal. General infosec knowledge supports practitioners, when building a

security architecture or responding to an ongoing intrusion, because general knowledge indicates courses of action that will plausibly be successful. The boundary between general knowledge and case-specific knowledge is not perfectly sharp. Both CISOs (chief information security officers) at big companies and malware analysts are practitioners who use general knowledge, but what counts as general knowledge to the malware analyst probably seems very case-specific to the CISO. Nevertheless, the field as a whole knows rather a lot.

5 Conclusion

Returning to the philosophical questions with which we began, it should by now be clear that there are nothing like general laws in infosec, and far from anything like a scientific theory in the traditional sense. General knowledge in infosec is not gained by finding general laws or theories. General knowledge is as hard to win in infosec as it is anywhere, due to the triad of challenges. Each of these—changeability of software, active adversaries, and justified secrecy—alone could frustrate generality.

Yet the case is not hopeless; infosec has seen progress, however fitful. We began with the initial strategy of looking to mechanism discovery from philosophy of science to illuminate this, and we have shown that it is fruitful. We can see general knowledge built up by discovering clusters of related multifield mechanism schemas, that we should expect to vary along four dimensions: activities and entities, phenomena, organization, and etiology. We demonstrated how this can help us conceptualize work in infosec by studying in detail three cases of active research problems within infosec that can be seen as discovering—and sharing—mechanisms. The cases of mechanisms we discuss in fact vary along these dimensions. Although each infosec case comes from a different field of expertise, and each focuses on a different element of mechanisms, the three cases are nevertheless strongly interdependent. Our approach also happens to indicate something about what makes information security a coherent field. Just as Craver (2007) describes the "mosaic unity" of neuroscience, built up by interleaving constraints applied by various fields on multifield mechanisms, we may describe a "mosaic unity" of infosec. There is little else that joins applied logicians verifying software correctness with criminologists interviewing malware authors into a coherent field of practice.

We could study many further interrelated cases; three examples can show us what is possible, but falls far short of defining a whole field. But what we have examined can provide the beginning of a framework for understanding the way general knowledge works for other problems within infosec. For example, we might apply such a framework to (in the language of mechanisms) other phenomena, such as skimming credit card details at point-of-sale devices. These are constrained by how the criminals monetize their gains, attack the point-of-sale device, and the malware's development history and how it hides itself. Practitioners build explanations and models in ways similar to the examples we have described when, for example, examining how the point-of-sale devices at the retailer Target, which has stores across the

USA, were leveraged to steal tens of millions of credit card details before Christmas 2013 (Krebs 2014).

These explanatory models of mechanisms are not developed in isolation. For example, the authors of the NCA's criminal business model would have been aware of an attack model similar to the kill chain, if not the kill chain itself. The kill chain constrains the hypothesized organization of the criminals' activities. Delivery happens before exploitation. This matches the criminal business model. And still higher-level mechanisms are also relevant. For example, mechanisms understood from the fields of internet architecture and international finance also constrain the criminal business model. Via examples like this criminal business model, one can see how international finance places constraints on the kill chain. Infosec practitioners have used such lessons to notice that, in some cases, the Visa payment system was the weakest point in a criminal's mechanism (Kanich et al. 2011). This example demonstrates one way in which constraints help lead to general knowledge. If an inter-field constraint applies to a part of a mechanism, and that mechanism is related to a second based on one of Glennan's similarities (in this case, the activity of criminal action on objectives via the kill chain), then other mechanisms similar along the same dimension to the same part of the mechanism may also be subject to the same constraint. Similarity among mechanisms provides a path for generalizing knowledge to other contexts.

There is another way in which the general knowledge of infosec is not like building a theory in the usual sense, such as building the theory of relativity—at least not as that has been traditionally treated in philosophy of science. The knowledge of infosec is very practically oriented, aimed at security, which is a very dynamic matter of constantly preventing and analyzing attacks. This practical aspect still needs more attention within philosophy of science. We wish to work further on integrating the ideas here with ideas developing the nature of evidence of mechanism in medicine (Clarke et al. 2014; Illari 2011) and the development of the important questions a philosophy of infosec should answer (Spring et al. 2017). Ultimately we will work on showing how a particular entity, activity, form of organization or etiology in one place may be well evidenced in infosec, and that evidence communicated effectively so that it may be made use of at other levels throughout the sociotechnical system we have described.

In conclusion, we have shown that even in the absence of laws, in a domain that is about as diverse and changeable as exists, and which has the special problem of secrecy, general shareable knowledge is still possible. This can be seen as the painstaking building of clusters of multifield mechanism schemas which vary along at least four dimensions of similarity: phenomena, activities and entities, organization, and etiology. Infosec provides cases in which practitioners successfully build general knowledge along these dimensions.

Acknowledgements The authors thank Inge de Bal, Giuseppe Primiero, Dingmar van Eck, and Stuart Glennan for fascinating conversations.

Stewart Garrick for comments and use of his GameOver Zeus diagram.

Spring is supported by University College London's Overseas Research Scholarship and Graduate Research Scholarship.

References

Addis, B., & Garrick, S. (2014). Botnet takedowns – our GameOver Zeus experience. In *Botconf, AILB-IBFA, Nancy, France*.

Alberts, C., Dorofee, A., Killcrece, G., Ruefle, R., Zajicek, M. (2004). Defining incident management processes for CSIRTS: A work in progress. Tech. Rep CMU/SEI-2004-TR-015, Software Engineering Institute, Carnegie Mellon University.

Andersen, H. (2017). What would Hume say? Regularities, laws, and mechanisms. In Glennan, S., & Illari, P. (Eds.) *Handbook of mechanisms and the mechanical philosophy*. London: Routledge.

Anderson, R.J. (2001). Why information security is hard: an economic perspective. In *Computer security applications conference, IEEE, New Orleans, LA* (pp. 358-365).

Anderson, R.J., & Moore, T. (2006). The economics of information security. *Sci.*, *314*(5799), 610–613.

Angius, N., & Tamburrini, G. (2017). Explaining engineered computing systems' behaviour: the role of abstraction and idealization. *Philos. Technol.*, *30*(2), 239–258.

Axelsson, S. (2000). The base-rate fallacy and the difficulty of intrusion detection. *ACM Trans. Inf. Syst. Secur. (TISSEC)*, *3*(3), 186–205.

Bartholomew, B., & Guerrero-Saade, J.A. (2016). Wave your false flags! deception tactics muddying attribution in targeted attacks. Tech. rep., Kaspersky Lab USA, Woburn, MA, presented at Virus Bulletin.

Bechtel, W. (2007). *Mental mechanisms: philosophical perspectives on cognitive neuroscience*, 1st. London: Routledge.

Bechtel, W., & Richardson, R.C. (1993). *Discovering complexity: decomposition and localization as strategies in scientific research*, 1st. Princeton: NJ.

Bogen, J., & Woodward, J. (1988). Saving the phenomena. *Philos. Rev. XCVII*, *3*, 303–352.

Boolos, G.S., Burgess, J.P., Jeffrey, R.C. (2002). *Computability and logic*, 4th. Cambridge: Cambridge University Press.

Brooks Jr, F.P. (1995). *The mythical man-month: essays on software engineering*, 2nd. Boston: Addison Wesley.

Caltagirone, S., Pendergast, A., Betz, C. (2013). The diamond model of intrusion analysis. Tech. rep., Center for Cyber Intelligence Analysis and Threat Research. http://www.threatconnect.com/methodology/diamond_model_of_intrusion_analysis.

Cartwright, N. (1983). *How the laws of physics lie*. Oxford: Clarendon Press.

Cartwright, N. (2012). *RCTs, evidence, and predicting policy effectiveness*, (pp. 298–318). Oxford: Oxford University Press.

Clarke, B., Gillies, D., Illari, P., Russo, F., Williamson, J. (2014). Mechanisms and the evidence hierarchy. *Topoi*, *33*(2), 339–360.

Craver, C. (2006). When mechanistic models explain. *Synthese*, *153*(3), 355–376.

Craver, C. (2007). *Explaining the brain: mechanisms and the mosaic of unity of neuroscience*. Oxford: Oxford University Press.

Craver, C., & Tabery, J. (2017). Mechanisms in science. In Zalta, E.N. (Ed.) *The stanford encyclopedia of philosophy, spring 2017 edn, Metaphysics Research Lab, Stanford University*.

CSEC (2013). Cyber security research and experimental development program. Tech rep., Communications Security Establishment Canada, Ottowa.

Darden, L. (2006). *Reasoning in biological discoveries: essays on mechanisms, interfield relations, and anomaly resolution*. Cambridge: Cambridge University Press.

Darden, L., & Craver, C. (2002). Strategies in the interfield discovery of the mechanism of protein synthesis. *Stud. Hist. Phil. Biol. Biomed. Sci.*, *33*(1), 1–28.

Darden, L., & Maull, N. (1977). Interfield theories. *Philos. of sci.*, *44*, 43–64.

Dupré, J. (2012). *Processes of life: essays in the philosophy of biology*. Oxford: Oxford University Press.

Elster, J. (1983). *Explaining technical change: a case study in the philosophy of science*. Cambridge: Cambridge Univ Press.

Elster, J. (1989). *Nuts and bolts for the social sciences*. Cambridge: Cambridge Univ Press.

Flechais, I., Riegelsberger, J., Sasse, M.A. (2005). Divide and conquer: the role of trust and assurance in the design of secure socio-technical systems. In *Workshop on new security paradigms, ACM, Lake Arrowhead, California, NSPW 33-41*.
Floridi, L., Fresco, N., Primiero, G. (2015). On malfunctioning software. *Synthese*, *192*(4), 1199–1220.
Friedman, M. (1974). Explanation and scientific understanding. *J. Philos.*, *71*(1), 5–19.
Galison, P. (2012). Augustinian and Manichaean science. Symposium on the Science of Security. National Harbor: CPS-VO. http://cps-vo.org/node/6418.
Gal-Or, E., & Ghose, A. (2005). The economic incentives for sharing security information. *Inf. Syst. Res.*, *16*(2), 186–208.
Glennan, S. (1997). Capacities, universality, and singularity. *Philos. Sci.*, *64*(4), 605–626.
Glennan, S. (2005). Modeling mechanisms. *Stud. Hist. Phil. Biol. Biomed. Sci.*, *36*(2), 443–464.
Glennan, S. (2010). Ephemeral mechanisms and historical explanation. *Erkenntnis*, *72*, 251–266.
Glennan, S. (2011). Singular and general causal relations: a mechanist perspective. In Illari, P., Russo, F., Williamson, J. (Eds.) *Causality in the sciences* (pp. 789-817). Oxford: Oxford University Press.
Glennan, S. (2017). *The new mechanical philosophy*. Oxford: Oxford University Press.
Glennan, S., & Illari, P. (2017). Mechanisms and the new mechanical philosophy. Routledge.
Grier, C., Ballard, L., Caballero, J., Chachra, N., Dietrich, C.J., Levchenko, K., Mavrommatis, P., McCoy, D., Nappa, A., Pitsillidis, A., Provos, N., Rafique, M.Z., Rajab, M.A., Rossow, C., Thomas, K., Paxson, V., Savage, S., Voelker, G.M. (2012). Manufacturing compromise: The emergence of exploit-as-a-service. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security, Raleigh, North Carolina, USA, CCS '12, pp 821–832*.
Hafner, K. (1998). Lyon m, Where wizards stay up late: the origins of the Internet. Simon and Schuster.
Hatleback, E., & Spring, J.M. (2014). Exploring a mechanistic approach to experimentation in computing. *Philos. Technol.*, *27*(3), 441–459.
Hempel, C.G. (1965). *Aspects of scientific explanation*. New York: Free Press.
Herley, C., & van Oorschot, P. (2017). Sok: Science, security, and the elusive goal of security as a scientific pursuit. In *Symposium on Security and Privacy (Oakland) IEEE, San Jose, CA*.
Hutchins, E.M., Cloppert, M.J., Amin, R.M. (2011). Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research*, *1*, 80.
Illari, P.M. (2011). Mechanistic evidence: disambiguating the Russo–Williamson thesis. *Int. Stud. Philos. Sci.*, *25*(2), 139–157.
Illari, P.M. (2013). Mechanistic explanation: integrating the ontic and epistemic. *Erkenntnis*, *78*, 237–255.
Illari, P., & Williamson, J. (2012). What is a mechanism? Thinking about mechanisms across the sciences. *Eur. J. Philos. Sci.*, *2*(1), 119–135.
Illari, P.M., & Williamson, J. (2013). In defense of activities. *Journal for General Philosophy of Science*, *44*(1), 69–83.
JASON Office (2010). Science of cyber-security. Tech. Rep. JSR-10-102 MITRE Corporation, McLean, VA.
Kaiser, M.I. (2011). The limits of reductionism in the life sciences. *Hist. Philos. Life Sci.*, *33*(4), 453–476.
Kanich, C., Weaver, N., McCoy, D., Halvorson, T., Kreibich, C., Levchenko, K., Paxson, V., Voelker, G., Savage, S. (2011). Show me the money: Characterizing spam-advertised revenue. In *20th USENIX Security Symposium, San Francisco, CA*.
Kincaid, H. (2011). Causal modelling, mechanism, and probability in epidemiology. In Illari, P., Russo, F., Williamson, J. (Eds.) *Causality in the sciences* (pp. 70-90). Oxford: Oxford University Press.
Kitcher, P. (1981). Explanatory unification. *Philos. Sci.*, *48*(4), 507–531.
Krebs, B. (2014). Target hackers broke in via hvac company. http://krebsonsecurity.com/2014/02/target-hackers-broke-in-via-hvac-company/, accessed Mar 2017.
Kührer, M., Rossow, C., Holz, T. (2014). Paint it black: evaluating the effectiveness of malware blacklists. Tech. Rep TR-HGI-2014-002, Ruhr-Universität Bochum, Horst Görtz Institute for IT Security.
Lawrence Livermore National Laboratory (2016). Rose compiler infrastructure. http://rosecompiler.org/.
Leonelli, S. (2009). Understanding in biology: the impure nature of biological knowledge. In De regt H.W., Leonelli, S., Eigner, K. (Eds.) *Scientific understanding: Philosophical perspectives* (pp. 189-209). Pittsburgh: University of Pittsburgh Press.
Lin, P.H., Liao, C., Quinlan, D.J., Guzik, S. (2015). Experiences of using the OpenMP accelerator model to port DOE stencil applications. In *11Th international workshop on openMP (IWOMP), Aachen, Germany* (pp. 45-59).
Lucas, J.r., R.E. (1976). Econometric policy evaluation: a critique. In *Carnegie-rochester conference series on public policy, elsevier*, (Vol. 1 pp. 19-46).
Machamer, P., Darden, L., Craver, C.F. (2000). Thinking about mechanisms. *Philos. sci.*, *67*, 1–25.

Metcalf, L.B., & Spring, J.M. (2015). Blacklist ecosystem analysis: spanning Jan 2012 to Jun 2014. In *The 2nd ACM workshop on information sharing and collaborative security, Denver, pp 13–22*.
Mitchell, S.D. (1997). Pragmatic laws. *Philos. Sci.*, *64*, S468–S479.
Mitchell, S.D. (2003). *Biological complexity and integrative pluralism*. Cambridge: Cambridge University Press.
Mitchell, S.D. (2009). *Unsimple truths: science, complexity, and policy*. Chicago: University of Chicago Press.
Moore, T., & Clayton, R. (2011). The impact of public information on phishing attack and defense. *Commun. Strateg.*, *81*, 45–68.
O'Meara, K., Shick, D., Spring, J.M., Stoner, E. (2016). *Malware capability development patterns respond to defenses: Two case studies. Tech. rep., Software Engineering Institute*. Pittsburgh: Carnegie Mellon University.
Piccinini, G. (2007). Computing mechanisms. *Philos. Sci.*, *74*(4), 501–526.
Radder, H. (2017). Which scientific knowledge is a common good? *Soc. Epistemol.*, *31*, 431–450.
Rapoport, A. (1966). *Two-person game theory: the essential ideas*. New York: Courier Dover Publications.
Sood, A.K., & Enbody, R.J. (2013). Crimeware-as-a-service: a survey of commoditized crimeware in the underground market. *Int. J. Crit. Infrastruct. Prot.*, *6*(1), 28–38.
Spring, J.M., & Hatleback, E. (2017). Thinking about intrusion kill chains as mechanisms. Journal of Cybersecurity 2(2).
Spring, J.M., Moore, T., Pym, D. (2017). Practicing a science of security: A philosophy of science perspective. In *New Security Paradigms Workshop, Islamorada, FL*.
SPSP (2017). Society for philosophy of science in practice: Mission statement. http://www.philosophy-science-practice.org/en/mission-statement/ accessed Jul 2017.
Steel, D. (2008). *Across the boundaries: Extrapolation in biology and social science*. Oxford: Oxford University Press.
Sundaramurthy, S.C., McHugh, J., Ou, X.S., Rajagopalan, S.R., Wesch, M. (2014). An anthropological approach to studying csirts. *IEEE Secur. Priv.*, *5*, 52–60.
Tedre, M. (2011). Computing as a science: a survey of competing viewpoints. *Mind. Mach.*, *21*(3), 361–387.
Tedre, M., & Moisseinen, N. (2014). Experiments in computing: a survey. The Scientific World Journal.
Tempini, N., & Leonelli, S. (2018). Concealment and discovery: the role of information security in biomedical data re-use. Social Studies of Science In press.
Thompson, K. (1984). Reflections on trusting trust. *Commun. of the ACM*, *27*(8), 761–763.
Turing, A.M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *J. of Math.*, *58*(345-363), 5.
University College London (2017). The research institute in science of cyber security (riscs). https://www.riscs.org.uk/, accessed Mar 6, 2017.
Winskel, G. (1993). *The formal semantics of programming languages: an introduction*. Cambridge: MIT Press.
Woodward, J. (2003). *Making things happen: a theory of causal explanation*. Oxford: Oxford University Press.
Yakdan, K., Dechand, S., Gerhards-Padilla, E., Smith, M. (2016). Helping Johnny to analyze malware. In *IEEE Security & Privacy (Oakland), San Jose, CA*.

Affiliations

Jonathan M. Spring[1] · **Phyllis Illari**[2]

Phyllis Illari
phyllis.illari@ucl.ac.uk

[1] Computer Science, University College London, London, WC1E 6BT, England

[2] Science and Technology Studies, University College London, London, WC1E 6BT, England