JOURNAL OF CYBERSECURITY

Journal of Cybersecurity, 3(3), 2017, 185–197
doi: 10.1093/cybsec/tyw012
Advance Access Publication Date: 4 January 2017
Research paper

Research paper

Thinking about intrusion kill chains as mechanisms

Jonathan M. Spring,[1,3,*] and Eric Hatleback[1,2]

[1]Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA; [2]School of Computing and Information, University of Pittsburgh, Pittsburgh, PA 15260, USA and [3]Computer Science, University College London, London, WC1E 6BT, UK

*Correspondence address. E-mail: jspring@sei.cmu.edu

Received 22 November 2015; revised 1 July 2016; accepted 12 October 2016

Abstract

We integrate two established modeling methods from disparate fields: mechanisms from the philosophy of science literature and intrusion kill chain modeling from the computer security literature. The result demonstrates that model accuracy can be improved by incorporating methods from philosophy of science. Modeling security accurately is a key function in the science of security. Mechanistic modeling of computer security incidents clarifies the existing model and points toward areas for substantive improvement for computer security professionals. Additional models of computer security incidents are translated mechanistically to compare results and to demonstrate such modeling can be applied in multiple situations. This integration of philosophy of science and computer security is sensible only by integrating new adaptations to mechanistic modeling, specifically conceived to enable better modeling of engineered systems such as computers. The results indicate continued integration of the fields of philosophy of science and information security will be fruitful.

Key words: modeling; investigative strategy; incident response; mechanisms; systems modeling; adversary modeling

Introduction

Models are important to science and to rational inquiry more generally. In particular, "scientists use models to represent aspects of the world for various purposes. On this view, it is models that are the primary . . . representational tools in the sciences" [1,p.747]. The purposefulness of modeling is critically important. Computer Network Operations (CNO) is the general term that encompasses attack, defense, and exploitation using computer networks [2]. Our purpose in modeling CNO by incorporating mechanistic thinking is to understand intrusions more thoroughly and, ultimately, to reduce the damage and disruption caused by CNO. Having such a model enables both better incident response (IR) and better computer network defense (CND) because responders and defenders can more adeptly understand the situation by interpreting it via the model, despite the occasional oversimplification. Network analysts may not think of themselves as scientists, but they share goals such as finding explanations and causes. As Hatleback and Spring [3] argued previously, analysts stand to benefit from adopting techniques honed by scientists.

Our purpose is critically important in our modeling language choice. For example, it may seem more natural to use a modeling language like Unified Modeling Language (UML) since it is a common model to design computer systems [4]. UML and other software engineering models are not incompatible with scientific modeling via mechanisms, although work codifying an engineered mechanism is nascent [3]. However, contrary to a systems engineer, yet like a scientist, the security practitioner attempting to understand an incident must build a model that includes physical, human, and engineered elements. Also like a scientist, the security analyst must form a descriptive model of how the world is working, unlike an engineer, i.e. a designer, whose model goal is to satisfice particular desired features of the design [5, p. 119ff]. Therefore, we choose to adopt the language of contemporary scientific modeling rather than software design, as the goals of the scientist more closely align with those of the computer security analyst.

We propose to enrich the kill chain model [6] by integrating well-developed modeling ideas from the literature on mechanisms in

the philosophy of science literature. We do not claim that the kill chain model is complete or perfect. Nonetheless, it is a common model of CNO with enough detail to provide an instructive starting point. Work knitting together security and philosophy is still young. However, such work offers the opportunity to make the existing modeling in security more robust.

The mechanisms' literature in philosophy of science includes areas such as what counts as a useful explanation and how to design studies to arrive at explanations. For concreteness, we adopt the consensus definition that "a mechanism for a phenomenon consists of entities and activities organized in such a way that they are responsible for the phenomenon" [7, p. 120]. However, as a practical matter, we are more interested in the threads of the literature related to constructing useful explanations and designing our observations to better discover mechanisms than more abstract threads. For example, Machamer et al. [8, p. 2] introduce their seminal work with "mechanisms are sought to explain how a phenomenon comes about or how some significant process works." Similarly, responders and defenders try to explain an attack or discern how it works, in the effort to prevent future attacks. Philosophers working on mechanistic explanation or description often see themselves as providing an account of how competent scientists solve complex problems and learn about the world [9, p. 7]. One of our goals is to adapt this advice for competent problem-solving to the security domain, in order to judge where it may be helpful.

Prior work in security is largely compatible with mechanistic modeling. Alberts *et al.* [10] identify five high-level processes within incident management: prepare, protect, detect, triage, and respond. Within each, the defender must have knowledge of the available mechanisms (mechanisms to protect, mechanisms to detect, etc.) as well as the likely mechanisms of attack. We focus on CNO modeling as an example application because it provides this touch point into many other security processes. Consistent translation across disparate specialties into a common parlance is an advantage of mechanistic modeling in the sciences generally, and security work should likewise benefit. For example, Basin *et al.* [11] incorporate physical properties into security models; Liu *et al.* [12] incorporated attacker objectives in network defense modeling to improve defense. By integrating the modeling language of mechanisms from philosophy of science, such varied research can be better integrated. A model of a particular attack will be used differently by those implementing detection technologies, which are forward-looking, than those who implement response plans, which are backward-looking. But both detection and response will want useful explanations of an attack, and to be confident their explanations are consistent with each other. The mechanisms' literature offers advice for these tasks.

Mechanisms are comprised of entities and activities [8]. We describe the seven elements of the intrusion kill chain as activities. This approach permits the CND analyst to think about the process in a structured level of detail. We further propose that proper defenses involve entities, modeled at the level of detail of the kill chain. This modeling choice presents a coherent picture of the devices used by adversaries and defenders as entities. This insight makes two contributions: one specific to the kill chain and one to the security community broadly. The specific contribution of thinking about this process mechanistically is importing the detailed modeling and design of scientific observations to enrich our CND and IR understanding of adversary CNO. The broader contribution is to introduce CND thinking to mechanistic modeling, thereby improving communication among network security professionals in different specializations. The purpose of both contributions is to enable better network defense by operators, analysts, researchers, and scientists.

The methodological differences between forensic cybersecurity and descriptive cybersecurity will need to be explored. This distinction has been elucidated for other disciplines with a forensic and traditional science subfields, such as geological science [13]. Both fields benefit from adopting better modeling from modern science because even though the methods diverge, the modeling language and jargon used in forensic versus descriptive science within a field often is quite similar since the domain of expertise is the same. The model of attacks used as an example in the following sections is targeted at forensic cybersecurity analysts as they investigate an incident, and so this subfield is the primary focus of this article. An example of descriptive cybersecurity science is investigating the existing blacklist ecosystem in order to understand and contextualize the many other activities or investigations that use blacklists, as done by Metcalf and Spring [14]. Both descriptive and forensic cybersecurity stand to benefit from integrating mechanistic modeling, however, we take the example of forensic cybersecurity because existing work in that subfield is richer. There is a further synergistic benefit to both descriptive and forensic cybersecurity in that if both can speak the same modeling language, namely mechanisms, then the communication between the two subfields will be easier and faster, hence advances in one can be more quickly adopted and impact the other.

There are various tasks in security which are at heart problem-solving tasks to which our project is applicable. For concreteness, consider the exploitation mechanism of a drive-by download, an example we use later in Fig. 4. A practitioner doing each activity, to prepare, to protect, to detect, to triage, and to respond [10] will emphasize different questions given the mechanism for a drive-by download. A detector may ask what aspects of the mechanism distinguish it from benign traffic. A preparer may ask how commonly, and what sorts of, adversaries tend to use drive-by downloads to determine resource allocation to the protectors and detectors. And so on. This example highlights an important feature of mechanistic explanation: mechanisms are composable and arranged in a loose hierarchy of levels. We select examples throughout to demonstrate the composability of mechanisms in security. These features are prominent in the success of mechanistic explanation in neuroscience [15], as we introduce in Section 3. Both the detector and preparer naturally take advantage of this feature; the detector asks about lower-level details that are distinguishing, and the preparer asks about higher-level details of actors in the way that Caltagirone *et al.* [16] does. This example also highlights the way in which a single mechanism is not a silver-bullet explanation, but rather contributes to a network of improved understanding.

The intrusion kill chain model that is introduced in Section 2 is not the only formulation of attack steps or phases. For example, Bejtlich uses "Reconnaissance, Exploitation, Reinforcement, Consolidation, Pillage." Howard and Longstaff [18, p. 16] codified a "computer and network incident taxonomy" that also has seven major parts: Attackers, Tool, Vulnerability, Action, Target, Unauthorized Result, and Objectives. Much of our application of mechanistic modeling is equally applicable to other descriptions, as we demonstrate in Section 6. We demonstrate mechanistic modeling using the kill chain model; but a robust mechanistic representation of computer network security incidents would also incorporate other sources. When we translate models into mechanistic models we often find explanatory gaps. Mechanistic models are well suited to assist the investigator in finding such gaps, which is one argument in favor of using them. We propose refinements to both Hutchins

et al. [6] and Howard and Longstaff [18] that bridge these explanatory gaps and provide a more robust model of CNO.

We begin with a summary of the relevant seminal papers: Hutchins *et al.* [6] on intrusion kill chains in Section 2; and Machamer et al. [8] on mechanisms in Section 3. Section 4 provides motivation for knitting the models together. In Section 5, we move forward with unifying the two models. Section 6 provides an example of incorporating the kill chain model of attacks into more coarsely-grained models of CNO incidents, and Section 7 presents an example of incorporating more finely-grained mechanisms to detail a specific activity of the kill chain; together these examples demonstrate the connective explanatory power of mechanistic modeling. Section 8 discusses adding probabilistic details to mechanistic models. Section 9 outlines some foreseeable difficulties and references existing solutions. In Section 10, we offer possibilities for future work and concluding remarks.

Kill chains

In the intrusion kill chain model, adversaries execute intrusions in discernible stages or steps. The seven defined stages provide the incident responder or CND architect with a framework for reasoning about intrusions. It is best to detect an intrusion at the earliest possible stage. The model provides some clarity with respect to defining "early," namely as fewer stages have been completed, and it also provides ready advice for responding to and measuring activity at each stage. The stages defined by Hutchins *et al.* [6, pp. 4–5] are as follows:

> **Reconnaissance** – Research, identification and selection of targets, often represented as crawling Internet websites such as conference proceedings and mailing lists for email addresses, social relationships, or information on specific technologies.
>
> **Weaponization** – Coupling a remote access trojan with an exploit into a deliverable payload,... Increasingly, client application data files such as Adobe Portable Document Format (PDF) or Microsoft Office documents serve as the weaponized deliverable.
>
> **Delivery** – Transmission of the weapon to the targeted environment. [Three common] delivery vectors for weaponized payloads... are email attachments, websites, and USB removable media.
>
> **Exploitation** – After the weapon is delivered to victim host, exploitation triggers intruders' code. Most often, exploitation targets an application or operating system vulnerability, but it could also more simply exploit the users themselves or leverage an operating system feature that auto-executes code.
>
> **Installation** – Installation of a remote access trojan or backdoor on the victim system allows the adversary to maintain persistence inside the environment.
>
> **Command and Control (C2)** – Typically, compromised hosts must beacon outbound to an Internet controller server to establish a C2 channel.... Once the C2 channel establishes, intruders have "hands on the keyboard" access inside the target environment.
>
> **Actions on Objectives** – Only now, after progressing through the first six phases, can intruders take actions to achieve their original objectives. Typically, this objective is data exfiltration ...; violations of data integrity or availability are potential objectives as well. Alternatively, the intruders may only desire access to the initial victim box for use as a hop point to compromise additional systems and move laterally inside the network.

These seven stages can occur simultaneously for different attacks, but they cover the set of common actions in a single attack.

Mechanisms

Mechanisms offer a way of formalizing how humans think about how the world works through our scientific investigation. Mechanisms are the prevailing way in which scientists describe what it is that they are investigating, though there are some less common alternatives. Philosophers of science provide the primary literature that tracks the methodology of scientists. In that literature, the mechanistic approach has gained favor over the previously dominant logical empiricist tradition. See Glennan [19], and the references therein, for an introductory account of the ascendance of the mechanistic approach.

We apply the mechanistic approach to computer security because it is effective. Fortuitously, the popularity of mechanistic thinking in other scientific disciplines means there is a wealth of existing work on how to use it to improve research. Using a similar structure in computer security would make it easier to make use of this existing literature.

The canonical account of mechanisms describes them as entities and activities organized to explain changes [8, p. 3]. Importantly, when a mechanism accurately describes a phenomenon, it permits the prediction of other events [8].

As introduced earlier, the modern consensus definition is that "a mechanism for a phenomenon consists of entities and activities organized in such a way that they are responsible for the phenomenon" [7, p. 120]. Understanding the mechanism helps us explain the phenomenon, predict it, and, in some cases, change it or prevent it from occurring [1]. In this regard, security may be like medicine. In medicine, statistical evidence is important, but explanations via mechanisms "help to determine whether positive results of a trial are due to genuine effectiveness or are simply a statistical blip; [mechanistic] evidence is also crucial when designing and interpreting a statistical trial, and when determining effectiveness in a new population or a particular patient" [20]. These considerations also apply in medicine when searching for a mechanism of cure when only knowing the mechanism of disease. It is with analogous goals and applications in mind that we seek to apply a mechanistic approach to cyber intrusions.

Illari and Williamson [7] elaborate on what it means to be responsible for a phenomenon, what counts as an entity or activity, and what qualifies as sufficiently organized. "Responsible" captures the diverse capacity of mechanisms in various tasks while maintaining an appropriate form of productive regularity or stability. Entities and activities are the result of decomposing an explanation into two kinds of parts: physical bits and pieces (entities), and what they do (activities).

Using "entity" and "activity" for these two kinds of parts of a mechanism have rhetorical and mechanism-discovery benefits over, say, considering activities to be merely properties of entities. For example, in protein synthesis, the activity is relatively uninteresting but the structural detail of the different entities matters a great deal. On the other hand, in "systems biology explanations, the entities are relatively similar to each other and the activities are vital to produce the phenomenon" [7, p. 126]. In this preferred language, the mapping of entities to activities is unconstrained. Compare the term activity to "capacities," which implies a unary mapping (one entity per capacity), or "interaction," which implies a binary mapping (at least two entities per interaction). Activity is explicitly chosen in order to be agnostic to the arity of the entity–activity organizational mapping.

Organization is quite broadly the relations among entities and activities, initial conditions, and ongoing conditions that allow the

phenomenon to be produced [7, p. 128]. Furthermore, organization relates a mechanism as a whole to its components. Components may be organized via relations of space (location, size, shape, motion), time (order, rate, duration), or activity (e.g. feedback) [15, p. 189]. Specifying organization is usually an empirical task, with different successful strategies in different fields.

Consider the mechanism for eating a sandwich. There are entities — the sandwich itself, teeth, lips, tongue, saliva — and activities — biting, chewing, swallowing. To explain how to eat a sandwich, one must organize these in the sensible way. Any haphazard combination of these activities is not likely to work; imagine if swallowing came before biting, for example. If all the entities are not present, most commonly teeth, the mechanism is also unlikely to proceed as expected (unless a suitable replacement is found). In this example, we see that mechanistic models may be used to explain a process and/or to diagnose the source of malfunction.

As another simple example, take the mechanism that captures the phenomenon of a seaside cliff eroding. There are entities — the cliff itself, sand particles, wind, waves, stones — and activities — colliding, blowing, churning. To explain how the cliff erodes, one must organize these properly. Any haphazard combination of these activities is not likely to work; colliding (i.e., of sand particles and stones with the cliff) cannot occur before the blowing of the wind or the churning of the waves, for example, because it is the very activity of blowing and churning that enable the activity of colliding. The elements of the mechanism provide good candidates for more careful measurement, if we need to estimate rates of erosion for insurance purposes, for example. However, without the qualitative mechanistic explanation, one would not know what aspects are relevant to measure.

Mechanistic models span multiple levels, which is to say that each model has a different scope and granularity of detail. The choice of level of explanation is roughly the trade-off between breadth and detail that every model must make to remain useful. A map on a scale of one kilometer per kilometer has no purpose; as Lewis Carroll snarks about using such a thing, "the farmers objected: they said it would cover the whole country, and shut out the sunlight!" [21]. Appropriate explanation is not just choosing the appropriate breadth and detail for the purpose at hand, but also selecting the relevant items and providing appropriate linkages both to broader and to more detailed mechanisms [15, p. 10]. With appropriate linkages, it is possible to acquire the appropriate granularity for any task, to put a complex task in context, and to prioritize candidates for influencing a complex task. By providing a common language for these linkages, the mechanistic account improves communication and comprehension.

Craver [15, p. 1] notes that "explanations in neuroscience describe mechanisms, span multiple levels, and integrate multiple fields." We contend that information security is analogous to neuroscience here. Craver works an extended example of the explanation of spatial memory in neuroscience to capture important points about mechanisms [15, ch. 5]. There are four levels of mechanisms for spatial memory. The level of spatial memory as such is studied by experimental psychologists testing rats learning to run mazes. The level of spatial map formation is studied by physiologists manipulating the computational properties of brain regions with drugs or scalpel e.g. to localize spatial map formation to the hippocampus. The cellular-electrophysiological level is studied by neurobiologist to identify the synaptic relations among neurons that contribute to the storage of spatial memory. Finally, the molecular level is studied by biochemists to determine the chemical and electrical composition of nerve cells that make such synaptic interaction possible. There are many potential methods to define "level" in this case. Craver argues against seven other methods (products, units, causation, size, part-whole, aggregation, and spatial containment) before concluding that the only suitable meaning is level of mechanism, in the sense that a lower-level mechanism is an entity and/or activity in the higher-level mechanism. Craver [15, p. 228] further argues that the unity of the field of neuroscience can only be properly understood when we see it as "different fields integrat[ing] their research by adding constraints on multilevel mechanistic explanations." Information security is likewise multidisciplinary, which feeds our motivation that the analogy usefully holds.

Mechanistic modeling is creating and refining a model of phenomena in question by so organizing the appropriate (models of) entities and activities. A computer scientist may initially conceptualize a mechanistic model as a sequence diagram, per UML, in that sequence diagrams also feature a set of entities and the activities between them. However, mechanistic models are not, and should not be, so strictly defined: mechanistic models must account for both engineered and physical mechanisms, where "engineered mechanisms are susceptible to having their entities or their activities changed during the course of the investigation at the will of a rational decision-making entity" [3, p. 445]. Physical mechanisms include astrophysics as well as neuroscience: one cannot consciously stop neurons firing to form spatial memories any more than consciously will a supernova. Information security and forensics also need to account for the general uncertainty of human investigation of the physical world. The practice of mechanistic modeling is a general account of what good models in the sciences have in common. There is not a more formal definition than the above from Illari and Williamson [7]. Yet, using this definition, mechanistic modeling can be general enough to capture all of a scientist's or security analyst's information about a phenomenon, provide guidance on what types of experiments and observations to prioritize based on a current model, and provide structure to what features we expect in a good model.

Why knit these together?

Models are important; they play many roles. In quantitative analysis, models such as mechanistic models serve many purposes that are needed in computer-network security work. Good models help the investigator estimate effects, measure where more precision is needed, impute or fill in missing data, and design tests to determine causation [22]. Equally important, models, and specifically mechanistic models, serve a critical purpose in enabling and improving communication among professionals in a complex field. Bechtel and Richardson [9] argue persuasively that scientists explain phenomena mechanistically at least in part because it is an effective search heuristic. The two main strategies are decomposition of the explanatory task into manageable subcomponents and localization of activities to specific subcomponents. This approach is coarse, and may fail, but failures are instructive and inform a revised subsequent attempt [9, pp. 23–24]. The literature on mechanisms is where we find the broad discussion of how to improve discovery and communication of complex concepts. This literature can help the diverse group of researchers, policy makers, and operators to move security productively toward "a story constrained by all the empirical contact with the world that ingenuity can design; a story that we can understand, manipulate and communicate, that we can use, and use collaboratively, to help us manipulate, control and predict the world—and lead science to better knowledge" [23, p. 253]. This project rings

true in security as clearly as anywhere. Security has a unique constellation of challenges to which we need to adapt mechanistic modeling.

Before moving forward, we shall defuse the argument that the constellation of challenges in security is unique and other scientific practices are inapplicable. If this were so, perhaps lessons from other sciences such as mechanistic modeling would be utterly unhelpful. That the constellation of challenges to security is unique is given, but unmoving; a unique set of challenges is a defining characteristic of any scientific field. Foremost, (i) computer security includes active adversaries that respond to defender actions; however, economics and game theory address this same adversarial nature, as does diagnostic psychiatry. Another common challenge is (ii) the rapidly changing targets that security must discover; yet pathogens studied by virologists evolve to bypass vaccines on the order of months. A further challenge is (iii) that computer security studies engineered artifacts, rather than natural systems; archeology, anthropology, and forensic criminologists all study engineered artifacts using scientific methods. In computing generally, we have (iv) the rapid changeability of software; to account for this Hatleback and Spring [3] argue for heuristically separating engineered from physical mechanisms, as defined in Section 3. But this separation is to enable clearer communication with and integration of scientific practice from other disciplines wherever possible, not to deter it. Another challenge is (v) the detection of rare events and managing the base-rate fallacy [24], but safety analysis of analog systems like nuclear power plants have similar challenges. Finally, security faces challenges of (vi) perverse economic incentives, such as moral hazard and tragedy of the commons [25], similar to challenges in public policy and economics.

With these six challenges, computer security is certainly diverse and difficult. However, as our examples show, computer security does not face any challenge that some other science does not face as well. Therefore, it is reasonable to attempt to fuse the mechanistic modeling approach employed in many other sciences with security. Adversaries will of course intentionally attempt to subvert our designed mechanism discovery tools and processes. Some mechanisms will be easy to change. But even with engineered mechanisms, such as the TCP/IP suite, there are certain tasks the adversary must do to successfully communicate. That mechanisms are changeable does not make them useless to our understanding and planning. Relatively changeability may, and should, inform how we target our studies and interventions. For example, Spring [26] suggests domain-name takedown strategy based on the adversary's relative ability to change various features of the Domain Name System (DNS).

Mechanistic modeling of CNO will help investigators make better predictions via assistance with quantitative and qualitative representation. A model represents what the investigator believes happens or happened in the world; in philosophical jargon, it represents the phenomenon [27]. Scientists and computer forensics analysts are the same in that both are investigators attempting to build accurate and warranted models. Mechanistic models of phenomena can readily have (subjective) probabilities assigned to whether an activity will proceed in one way or another, although in our examples below we simplify the diagrams by omitting probabilities. A mechanism may model using just a causal Bayes net or a graphical causal model. For example, the mechanisms of human cognition and learning are explained well as a graphical causal model [28].

A CNO investigator often needs to make decisions with limited resources. Time and/or information may be limited. For example, during a DDoS (distributed denial of service) attack, the incident responder has little time to decide if the DDoS is a distraction to cover for a more serious unauthorized access and take protective measures. This process is a different kind of limited resource than a forensics examiner making a retrospective damage and exfiltration assessment, where information of the initial infection vector has been rolled off logs if captured at all. Both use mechanistic explanations as evidence in their decisions. Both also use other information in their decisions, such as risk tolerances and their prior beliefs about what is most likely or common.

The example of the incident responder already implicitly includes mechanistic explanation in the decision-making. The responder knows that one mechanism adversaries use to achieve objectives is a series of attacks, one noisy and harmless and one quiet and damaging, to distract defenders from the true harm. If responders know this is a possible mechanism, they are more likely to make decisions and direct their observations to seek evidence of this second, covert attack. This particular attack mechanism is not as useful to the forensic examiner, who cares more about what valuable information was stolen than how the adversary got in. In this case, knowledge of possible mechanisms of data hiding and exfiltration will guide the search for evidence of harm, such as steganography, HTTP-over-DNS, peer-to-peer botnets, etc. In both cases, as described earlier in the case of medicine, evidence of possible mechanisms is crucial when designing and allocating resources to future observations, trials, studies, and investigations. The details change with the context. We explore the example of CNO because it applies to a wide variety of security contexts, albeit in different ways as demonstrated by the example contrasting the incident responder and the forensics examiner. In practice, this should be complemented with knowledge of mechanisms of defense, such as firewalls, IDS, access controls, network partitioning, encryption, and so on. As in medicine, knowing how a patient becomes sick is valuable to knowing how to cure them, but it is not the same as knowing how to cure them. By integrating information security with the existing mechanisms literature, such lessons as these from medicine can be brought to bear on analogous problems in security.

Utilizing a mechanistic model will assist a CNO investigator to assign subjective probabilities to capture the investigator's beliefs on what can happen or has happened. What kind of model the investigator will build depends on her goal; e.g. a CND goal will aim to mitigate the vulnerabilities that allowed the adversary in, whereas law enforcement may have a goal of attributing the attack to an agent and apprehending him or her. Current best practices in CNO attribution is summarized in the diamond model by Caltagirone *et al.* [16], while best incident management and recovery practices are well summarized in Alberts *et al.* [10] and Shimeall and Spring [29, ch. 15]. None of these approaches provide a method for incorporating or mitigating uncertainty of the investigator's beliefs directly in the model. Caltagirone *et al.* [16, p. 54] recognize their model is "cognitive and highly manual" and leave as future work improvements to reduce this problem; the ability to incorporate the formal modeling of causal Bayes nets via mechanistic modeling is one such improvement. The CND models do not explicitly call for this improvement as future work, but stand to benefit in the same way. Section 8 sketches how to integrate subjective probability into the relevant models.

Biology provides a good example of effective use of models for improving professional communication among diverse fields. The kill chain is similar to a model that a primary care physician might use to assess a single patient; it models CNO at a granularity of one system. However, different problems and different purposes require mechanisms at different levels of granularity. The mechanisms useful for a patient's day-to-day care do not help a molecular biologist

understand the mechanism by which poison ivy makes a rash; that is a different level of granularity. Likewise, an epidemiologist operates with a different, more coarse-grained, mechanistic view to understand disease within a population. All three perspectives of the world are simultaneously true and inform one another, but all three perspectives have different purposes. Due to human limitations, it is difficult for any one person to excel at understanding even one perspective in sufficient depth to advance it. Thus there are different professional fields, specialized essentially in different mechanistic granularities. The mechanistic model in these fields of science provides a crucial link to facilitate encapsulation of knowledge and professional communication.

Thinking about problems mechanistically improves communication among specialized individuals because it helps codify how the different perspectives interact with each other. A thorough modeling approach also helps identify when elements are not incorporated in the model; a mechanistic approach to modeling would help identify areas where an adversary could step around an existing security model to bypass it. We also argue that computer security would benefit from a more robust understanding of modeling because the field is now sufficiently complex that it surpasses the understanding of any single individual [30]. Importing structures already developed by philosophers of science, namely the mechanistic approach, would provide the needed modeling structure. We demonstrate this improvement by grafting the mechanistic modeling approach onto the intrusion kill chain model from the security literature [6].

Toward a unified model

The process of creating a unified model is at least as important as the model itself. As investigators and scientists gather more data, models change. But the process of refining a model changes much more slowly. We therefore describe not just the result, which always can be further refined, but the thought process by which we move toward a unified model. The kill chain is one of many possible starting points and is used here as much as an instructive starting point as a truthful model.

CNO is not the only aspect of cybersecurity that would benefit from mechanistic modeling. The Bell and LaPadula (BLP) model [31] for multilevel secure systems could cleanly be cast as mechanisms: subjects and objects are two types of entities, and activities are the classic actions initiated by subjects such as read and write. BLP then describes what set of mechanisms lead to the desirable behavior of a secure system. Incident management processes [10] are easily cast as mechanisms, with activities such as detect, triage, and mitigate. Cyber incident attribution [16] and cyber threat intelligence [32] also use models that could be translated to a mechanistic model. A model of CNO is a good starting place, however, because attacks are complex enough to provide interesting challenges while remaining tractable.

At a coarse level of granularity, a unified model is simple: cast the kill chain as a mechanism. The entity acting in each case is an "adversary." The activities are the seven steps provided in Section 2. These activities each require an entity as an object as well as an actor, and so at a coarse granularity, the object of the activities is the "target," or defender.

But this coarse-grained description abstracts away too much information to be useful. For example, the adversary and the target are not the only entities. There are finer-grained entities that are necessary to map the kill chain model accurately as a mechanism, such as "remote access trojan" ([33], see" backdoor" and" Trojan"), "exploit" [34], and "victim system" ([33], see" system component,"" system entity"). These entities have coherent definitions in the works cited, despite the fact that the kill chain paper does not define them or reference definitions [6]. Employing the mechanistic understanding helps make clear the relationship between these entities in the kill chain. Figure 1 provides a conceptualization of the seven steps in the kill chain mechanism.

The entities and activities in Fig. 1 are labeled with subscript *e* or *p* to indicate whether the element is engineered or physical, as defined in Hatleback and Spring [3]. Some elements have both engineered and physical components, subscripted *ep*, and these elements are good candidates for deeper exploration to tease apart which aspects are engineered and which are physical.

The activities, as arrows labeled 1 through 7 in Fig. 1, represent the seven steps of the kill chain provided in Section 2. The diagram provides a richer interaction describing the phenomenon. First, adversary *A* performs reconnaissance on target *T*. Second, the adversary weaponizes exploit delivery code *E*. Third, the exploit is delivered to the target. Fourth, weaponized code *E* exploits victim system *C*. Fifth, code *E* installs remote access code of some kind, *R*, on the victim system. Sixth, the remote code communicates over a command and control channel with the adversary. Seventh and lastly, the adversary completes actions on objectives against the victim system.

The activities labeled as having both an engineered and a physical component are "reconnaissance (1)," "command and control (6)," and "actions on objectives (7)." Reconnaissance is defined as

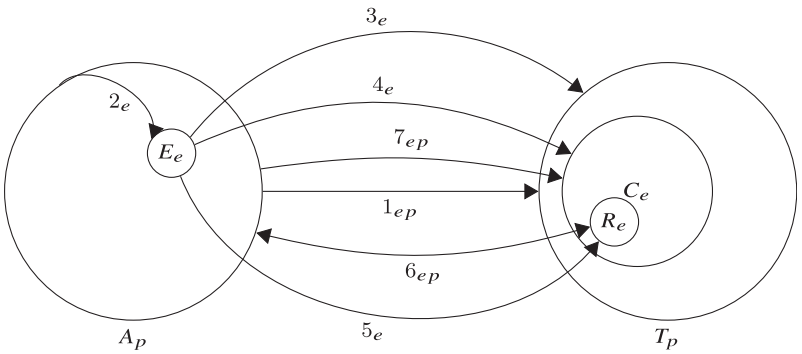

Figure 1. Rough mechanistic diagram relating the entities and activities involved in an intrusion, following the kill chain model of intrusions. The large entities are adversary *A* and target *T*; the medium-sized entity is target computer *C*; the small entities are weaponized exploit *E* and remote access trojan (RAT) or some malicious code *R*. The seven activities are labeled for the seven steps in the kill chain Hutchins *et al.* (6): reconnaissance (1), weaponization (2), delivery (3), exploitation (4), installation (5), command and control (6), actions on objectives (7).

"Research, identification, and selection of targets" [6, p. 4], where research and target selection are human activities with underlying physical mechanisms based in psychology or economics. Identification of targets is usually an engineered activity that requires scanners or other computer tools, such as the open-source Nmap [35]. Scanning also has limitations based in physics, such as the speed of light and bandwidth, which impact the volume of scanning that adversaries attempt.

Similar considerations are important for understanding "command and control" and "actions on objectives." Command and control may involve "hands on the keyboard" and human limits and choices, thus introducing physical mechanisms. Even if no direct human control is used, information theory limits control channels, information density, etc. (How elements such as algorithms should be considered with respect to the "engineered" versus "physical" distinction is not yet clear; this is an area of current work that impacts the engineered choices of the control channel.) "Actions on objectives" may mean data exfiltration, which is affected by physical mechanisms such as bandwidth limitations. However, if the actions on objectives include any effects on the physical world via cyber-physical systems, such as damage or disruption of services, physical mechanisms become quite relevant.

Constructing the model in this way makes clear both the areas in which we have more detail and the areas in which more detail is needed. A target likely has multiple types of systems, system components, etc., arranged in a system architecture. The target should know its own architecture well enough to list these entities; if it does not, it should prepare better, as recommended by (for example) CERT® incident management best practices [10]. One can imagine defining more fine-grained descriptions of what "exploit" means against each of these components. In some cases, this work is done and is simply not yet cast as a mechanism. Relevant work includes the common weakness enumeration (CWE) [36] and common vulnerability enumeration (CVE) [37].

Mechanistic modeling also helps identify what is not known. If the defender needs to complete an assessment of an incident, a mechanistic diagram of what 'could be' known compared to an assessment of what *is* known produces a list of important data items to research to better diagnose the intrusion. Perhaps the remote "command and control" and delivery [3] activities are known, but the adversary (*A*), exploit (*E*), and "actions on objectives" are not. Certain types of adversaries may use a certain control channel, knowing this activity helps identify the adversary and his or her potential goals. Although this line of reasoning is precisely the kind of disciplined thinking encouraged by the kill chain analysts [6,16], applying the mechanistic approach helps clarify which entity is doing what activity to which target and what the defender knows about those entities and activities.

A complete diagram of every possible computer network attack could not be readily comprehended by a human. Similarly, a maximally detailed view of how the human body works cannot be readily comprehended and used by a single doctor. Biological sciences have developed the capability to implement more fine- or coarse-grained mechanistic perspectives as needs require. If computer security and intrusion analysis likewise are to develop such a way of studying its complex system at different granularities as needed, the information must be organized to facilitate that goal. The mechanistic model has proven effective for other sciences to achieve this goal, and implementing such an approach in computer security may provide the means to achieve it. Casting important models, such as the kill chain model, as mechanisms is an important first step.

Different levels of granularity already are recognized in computer security. For example, there are large-scale network analysts, and host-based analysts. The Open Systems Interconnection (OSI) layers are a form of granularity levels with abstraction and encapsulation [38]. But the difficulty of communicating important information across levels of abstraction and among professionals with different specializations has not yet been overcome. By importing mechanistic thinking and attendant good scientific practices, we believe these communication deficiencies can be overcome.

A final benefit of unifying the two models involves the identification of areas for improvement. For example, delivery (3) is defined as "Transmission of the weapon to the targeted environment" whereas exploitation (4) is "after the weapon is delivered to victim host, exploitation triggers [malicious] code" [6, p. 4]. In Fig. 1, a gap exists between 3 and 4, since the target is different, though the entity acting appears to be the same. Thus, the approach permits us to question whether the definition of delivery is accurate or whether there is an additional activity describing how the weaponized code transits the target environment to get to the victim system. Gaps such as these are more easily identified when thinking mechanistically, since one goal in a complete mechanistic description is identifying explanatory gaps [8, p. 3].

We propose that delivery (3) is better understood as one of two activities, one for engineered targets and one for human targets. In a phishing email, there is a malicious link or file delivered to the human's email inbox. However, the exploitation (4) does not occur unless the human is tricked into opening the malicious content. For a clear example of delivery to a purely engineered target, consider the old ping-of-death vulnerability: as soon as the target computer received the malicious packet, it automatically processed it and crashed. Therefore, we propose that delivery_p, where a human is in the loop as the target, is a distinctly different activity than delivery_e, where a machine will be exploited automatically without human action. We call this human-centric delivery a physical mechanism because it depends on human psychology and vulnerability to trickery, which are physical mechanisms, despite the fact that the medium is electronic [3]. Delivery_p may have, but does not require, a delivery_e activity that occurs at the same time or immediately afterward. For example, if during delivery_p the human opens a malicious file, there is a delivery_e action that the file executes to exploit the human's machine. However, in some cases the human simply may divulge a password or other credential which then directly leads to exploitation by the adversary. In such a case, only the human, not a machine, has been tricked, and so no engineered delivery has occurred. In the case of such a delivery_p which only carries information back to the adversary and does not have a subsequent engineered delivery activity, there still appears to be an explanatory gap. We would model this as an information flow back to the adversary as part of the more detailed specification of the delivery activity; Fig. 2 hints at this with a bidirectional arrowhead on the line for delivery_p that is open, rather than filled black, to show it is present for only certain specifications of the activity. Figure 2 demonstrates this revised mechanistic diagram noting this branched path for delivery with dashed lines. The code to generate Figure 2 and Figure 4 is available in the Appendix.

This example of identifying a gap in the explanation warrants further exploration. One may ask how much the mechanistic approach to modeling has actually enabled this gap-finding, and how valuable finding gaps is. The model brings this gap to the fore quite naturally, not the subject-matter expertise needed to create the model. This fact is obvious because we have only translated the existing expert-created model into a mechanistic language. By doing this translation, the gap is readily apparent, when it was not before: there is no activity linking the effect of delivery of an exploit and the installation of the malicious code. Now that this is clear, we security

experts as scientists can posit what fits in that gap: the user clicking a link, bypassing a warning, viewing an email with a malicious font, or perhaps automated system action. While the content of the model and improving the model both require subject-matter expertise, translation into a mechanistic model, at the very least, provides a check on the explanatory soundness of our models by allowing ready inspection for gaps.

Incorporating other CNO models

Other CNO can be modeled in a way similar to what has been undertaken here with the kill chain. Some of these models will conflict with aspects of the kill chain. By translating multiple models into the same mechanistic language we can better compare them, extract the better parts, and implement the best aspects toward our unified model. The Bejtlich [17] model of "Reconnaissance, Exploitation, Reinforcement, Consolidation, Pillage" nearly matches the kill chain, except it is more coarse-grained. The attacker is assumed, not explicit, and the three steps in the kill chain of weaponization, delivery, and exploitation are subsumed into just "exploitation." "Reinforcement" is synonymous with "installation," and "pillage" is synonymous with "actions on objectives." "Command and control" and "consolidation" may seem to differ, but consolidation also means the control of compromised assets.

The Howard and Longstaff [18] model is more difficult to reconcile with the kill chain mechanistic model. This difficulty seems to arise because the Howard and Longstaff model involves "incidents," which are comprised of one or more attacks, whereas the kill chain models only single attacks. One potential solution is to model a single attack as a sub-mechanism within the larger mechanism of an "incident." Thus, the modeler might stipulate that there are two competing models for attacks, but that the model of incidents provided by Howard and Longstaff is unique. The modeler could make use of either model, but confusion could arise if the modeler does not note that what the kill chain calls "objectives" are what Howard and Longstaff call "unauthorized results," and what Howard and Longstaff call "objectives" are something else entirely: human objectives such as financial gain or glory.

The taxonomy of Howard and Longstaff already contains certain features of the mechanistic way of thinking. The taxonomy presents a temporal ordering of items for an incident, namely "Attackers -> Tool -> Vulnerability -> Action -> Target -> Unauthorized Result -> Objectives" [18, p. 16]. The taxonomy groups these items usefully; "events" are made up of just the "Action -> Target" sequence. "Attacks" stretch from the "tool" to the "unauthorized result." Thus, it already includes the flexible granularity of the mechanistic approach; an incident could also be modeled as "Attackers -> Attack -> Objectives" if a coarser granularity were appropriate.

However, the Howard and Longstaff common language for incidents can be clarified through the mechanistic approach by grouping the taxonomy into entities and activities, and clarified further by marking elements as engineered or physical. For example, an "Action" is an activity. Examples of actions listed are "probe, scan, flood, authenticate, bypass, spoof, read, copy, steal, modify, delete." In the context of computerized actions, these are all engineered activities. Preceding "Action" in the Howard and Longstaff model is "Vulnerability." A vulnerability is an entity that is usually associated with a larger entity within which the vulnerability is a flaw. The given classes of "vulnerability" are well-known in computer security: design, implementation, and configuration. Design vulnerabilities are best modeled as a physical entity, since they are based in mathematical, physical, or information-theoretic limitations that were not properly considered during design. Implementation and configuration vulnerabilities are engineered entities, since they only exist within the computer systems as implemented or configured incorrectly by humans.

The classification of items within the model can continue. "Attackers" are physical entities; "tools" are engineered entities. "Targets" are either physical or engineered entities. "Unauthorized results" are engineered activities and are generally enacted by the tools, which fit nicely with the mechanistic account that "entities are the things that engage in activities" [8, p. 3]. Finally, "Objectives" are a diverse category that includes physical entities (financial gain) and physical/engineered activities (damage).

Figure 3 presents a coarse-grained mechanistic visualization of Howard and Longstaff's common language for security incidents.

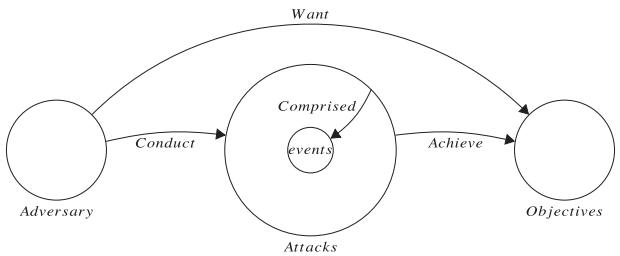

Figure 3. Visualization of the common language for computer security incidents Howard and Longstaff (18) as a mechanism, where "attack" is simplified to one entity that can be explored in many ways, with sub-mechanisms, at a finer granularity. One such sub-mechanism is described in Fig. 1.

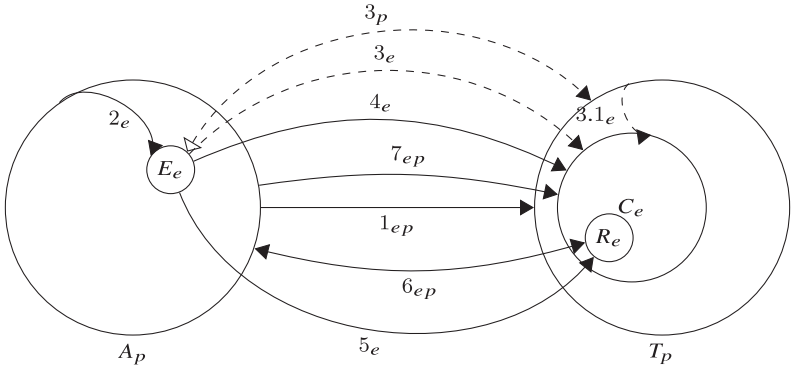

Figure 2. Improved kill chain mechanistic diagram, where delivery (3) in Fig. 1 is replaced by two options, an engineered activity directly to the target (3_e) and a physical activity through the human (3_p) with an optional secondary engineered delivery step (3.1_e).

Since there are more entities than activities specified, mechanistic models quickly identify some areas that need further detail. This identification is an important benefit of mechanistic modeling. For example, an adversary must "conduct" an attack. What that activity entails should be specified. Likewise, with respect to how "attacks" relate to "objectives," attacks "achieve" objectives, although exactly how this unfolds would benefit from further specification. Howard and Longstaff discuss "success and failure" of objectives related to whether the objective was achieved, but they provide no robust description of what this means. The mechanistic diagram highlights the importance of understanding this activity in order to understand the whole incident.

Mechanistic modeling also eases integration of other security research that can inform CND planning. For example, [12, Fig. 2] use a primitive kill chain-like model for attacks but graft a sophisticated game-theoretic model of attacker intent [12, p. 89ff] onto it. By seeing these two elements as separable mechanisms at different granularities, the analyst can upgrade the primitive attack model with the more appropriate elements from Howard and Longstaff [18] or Hutchins *et al.* [6]. At the same time, the attack models can benefit because of readier access to a more detailed model of attacker intent, which is currently absent therefrom. A game, in the formal sense, is a model that can also be cast as a mechanism with the players as entities and the details of each play as activities. A knowledge base (or information set) is a feature of each player-entity; payoffs or outcomes can be features of each activity. We do not suggest that every equation of game theory be rewritten as arrow diagrams. Rather, we offer this translation to explain how the insights from existing game-theoretic literature (see e.g. Alpcan and Başar [39]), using the correct criteria [40], can be integrated into mechanistic modeling in security.

On finer details

To be practically useful, a unified model will have to be able to explain real attacks and help investigations of them. Figure 4 models an obfuscated drive-by-download delivered via a malicious ad network, as described by Segura [41]. The mechanism is an example of the "delivery" activity from the kill chain examined at a finer granularity. This diagram focuses on the technical aspects of the delivery to the user's browser. The mechanism for how the ad networks select an ad for a user is left at a coarse grain, but could be modeled in more detail if more data becomes available [42]. Identifying such an item that requires more research because it is not clearly understood is a benefit of mechanistic modeling. Like science, Internet security is never done. But it helps to know where to go next just as much as it helps to know what is already well understood.

Most entities and activities in Fig. 4 have their common English meaning and the same modeling norms as Fig. 1 and Fig. 3. A special case is "fetches," which is transitive. That is, if the user's browser fetches a web page, which fetches an ad, which fetches a URL, the browser has made an HTTP request to fetch all three of these things. Modeling it this way preserves which resource redirected to which other resource, while an arrow from the browser to each resource would not.

It would be sensible to model the activities 3, 7, and 10 in Fig. 4 as yet finer-grained mechanisms. Activity 3 is target selection, which is the "reconnaissance" step in the kill chain, so we put it aside for now to focus on "delivery." Likewise, activity 10 indicates the end of "delivery" and the beginning of "exploitation." Therefore, consider activity 7, the calculation by which the JavaScript in the advert de-obfuscates the malicious URL. In this case, the mechanism is to use a regex to extract a string from the cookie and then unescape, split, and reverse that string, which yields a JavaScript HTTP request

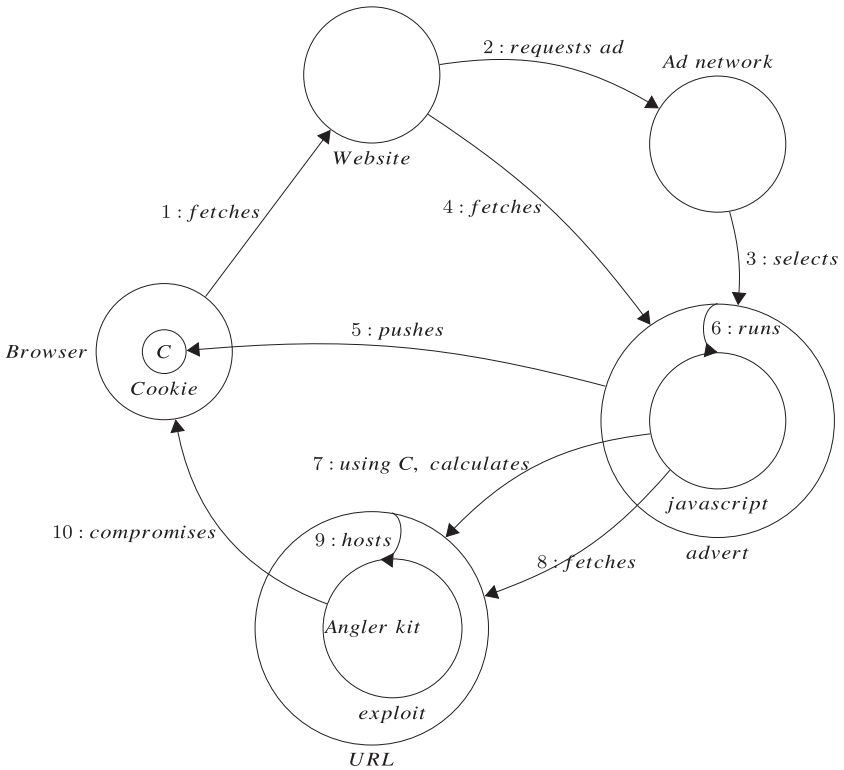

Figure 4. Diagram of an obfuscated drive-by-download delivered via a malicious web advertisement.

to a URL to be fetched (step 8). This mechanism defines a class of obfuscation techniques using cookies; the report by Malwarebytes [41] goes one level of granularity finer and specifies items like the specific cookie and regex used.

The creation of Fig. 4 demonstrates how we can use mechanistic modeling to move fluidly between modeling granularities to make sense of detailed cybersecurity events. Now our model can go from specific obfuscation techniques, to how such techniques are used in delivery of exploits via Fig. 4, to how delivery fits into attacks via the kill chain, to how attacks fit into adversary's human objectives via Howard and Longstaff. This flexibility greatly helps the researcher put fine-grained mechanisms into context, enumerate items to measure to counter a threat, and identify specific mechanisms—such as a URL obfuscation technique or ad delivery selection—for further investigation.

Including subjective probabilities

When a forensic investigator is actually trying to understand an attack, it is of course not so straightforward as a single path from adversary to unauthorized result. The investigator will believe different possible attack paths are more or less likely. We can model these beliefs in what phenomena are more or less likely using subjective probabilities [43]. The added detail from subjective probabilities increases the usefulness of mechanistic models. We call a mechanistic model an investigator holds, complete with subjective probability assignments, a "specified" mechanistic model, as opposed to a "general" mechanistic model which we have been discussing up to this point. The purpose of a general mechanistic model, and indeed the purpose of the kill chain or Howard and Longstaff's model of an intrusion, is to capture something useful about the pattern all, or at least nearly all, attacks follow. Therefore, following the kill chain, if an investigator observes an adversary successfully delivering malicious code to the defender's environment, then the kill chain model provides a good next guess as to what to look for or expect next, namely an exploit.

However, a practical network defender or investigator needs to know which exploit to look for in order to find or stop it. The information of "look for an exploit following a delivery event" restricts the scope of phenomena to look for, and so has value. But it is not sufficiently valuable to, by itself, allow CND work in practice. For this the CND investigator can readily use a specified mechanistic model; the added reasoning power of such a specified mechanistic model provides some obvious and large benefits over using the existing models, as existing models do not easily capture this reasoning on the investigator's belief about the actual phenomena perpetrated by the adversary.

Introducing the full spectrum of nuances included in the specified mechanistic model and how they circumvent the issues encountered with the general mechanistic model is beyond the scope of this article; we plan to describe them in detail as future work. But briefly we sketch the intuition and the shape of the way forward. Above, we mention that Howard and Longstaff [18] break down "action" to "probe, scan, flood,…" and so on. In the language of investigating CNO attribution from Caltagirone *et al.* [16], these actions are part of TTPs that we would expect different adversaries to use against different targets. So the investigator has a prior distribution on the set of actions for each type of adversary attacking each type of defender. In general, the defender is fixed as the investigator or the investigator's organization, so the distribution is what each type of adversary is likely to do. Types of adversary could be organized by capability level, similar to Spring *et al.* [44], to make the number of categories of adversary manageable. Many structured observations or experiments which do not immediately appear to bear on a mechanistic model actually are important to set better or more realistic priors on expected mechanisms; see e.g. Metcalf and Spring [14], Rasmussen and Aaron [45], and Kanich *et al.* [46].

Thus, practically, a CNO investigator benefits from mechanistic modeling because it provides a way to organize her current beliefs about adversary's likely paths, and to reason about what evidence would be necessary to support or condemn these hypothetical adversary paths during the course of the investigation. Figure 5 demonstrates an example notional prior distribution over the Howard and Longstaff [18] action category for three adversary types. This corresponds to an investigator being told, roughly, there was an unauthorized action on the system by one of three adversaries. This is not much of information, and thus there are lots of possible options. The matrix captures that the investigator believes there is a 15% chance that "$adversary_2$" will attempt and succeed at a "bypass" action. In general, Fig. 5 indicates scans are the most likely action, and so with no other data or context this potential action is the most rational subject for immediate investigation. Of course, a real practitioner knows the potential loss from a scan is very low, while data modification by an adversary can be quite costly to fix, and such considerations and background context would realistically figure in to the investigator's decision. The complexity uncovered by this brief sketch indicates why a full development of specified mechanisms is out of scope presently and requires future work.

One element highlighted by this discussion is the need for the investigator to set starting likelihood values that are at least sufficiently accurate for the purpose at hand; in statistics jargon, these values are called priors, in philosophy of science jargon, it is called realizing appropriate external validity. Whatever the term, the need is genuine.

Overcoming difficulties

The kill chain model, and computer science generally, has not yet adopted the modern mechanistic approach. Hatleback and Spring [3] describe two potential pitfalls: (i) mechanisms in nature behave regularly all or most of the time, whereas mechanisms engineered by humans, such as computer code, are changeable more easily; and (ii) there is inadequate existing literature in the philosophy of science that investigate the demonstration that activities exist, which is a problem because computer security focuses on searching out

$$\begin{bmatrix} probe \\ scan \\ flood \\ authenticate \\ bypass \\ spoof \\ read \\ copy \\ steal \\ modify \\ delete \end{bmatrix} \times \begin{bmatrix} adversary_1 \\ adversary_2 \\ adversary_3 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.9 & 0.25 \\ 0.6 & 0.8 & 0.96 \\ 0.001 & 0.9 & 0.11 \\ 0.5 & 0.5 & 0.2 \\ 0.6 & 0.15 & 0.2 \\ 0.01 & 0.6 & 0.3 \\ 0.6 & 0.2 & 0.9 \\ 0.1 & 0.7 & 0.95 \\ 0.65 & 0.5 & 0.99 \\ 0.75 & 0.05 & 0.05 \\ 0.01 & 0.92 & 0.0001 \end{bmatrix}$$

Figure 5. Notional adversary action prior probability distributions based on the Howard and Longstaff (18) classification of adversary actions. These priors are relative: the matrix does not sum to 1 in each column. If needed, they could easily be normalized to sum to 1.

unwanted activities. Both of these difficulties can be overcome, but it will take a concerted effort and dedicated research program.

Addressing the changeability of engineered mechanisms requires both philosophical and technical expertise. Nonetheless, both disciplines already have tools available to initiate the process.

In network security, there are devices that restrict communications and enforce security policies. An example policy is "only the web proxy can connect to web sites, and other hosts have to talk to the web proxy." This verbal policy would have a technical implementation. The policy, once implemented, restricts the entities and activities involved in communications. So, when thinking about what mechanisms to be concerned about in an intrusion, network security devices can help limit the set of mechanisms under consideration. If the security policy can be designed to reduce the arsenal of attack mechanisms available to the adversary, the changeability of the mechanism matters less because the plausible selection of attack mechanisms is reduced.

Unfortunately, computers today have a rather large attack surface [47], which means the possible attack mechanisms are many and the changeability of the engineered mechanisms matters. However, considering these attack vectors as mechanisms likely will help researchers identify common entities and activities, thereby prioritizing attention for solutions. Mechanisms in fields such as immunology already account for some amount of changeability, and so there should be some lessons available from, for example, experimental practice on quickly evolving microbes. Computer security has long sought to emulate certain aspects of immunology [48], and using the same scientific language of mechanisms should assist in identifying and transitioning relevant ideas between the two fields. Economics also seems a likely source of lessons learned for observations and modeling of a quickly changing system. Indeed, the economics of information security has had a dedicated conference for some time (http://econinfosec.org/weis-archive/).

The literature on demonstrating that activities exist may be sparse, but systems biology and computer security researchers do it already. More attention to the problem should contribute some current best practices in designing and reporting such observations. One difficulty in computer security is that these good examples often do not exist in traditional academic venues. Instead, they are found in industry reports and blog posts. Speed of reporting is often a higher priority than prestigious publication, due perhaps both to the youth of the field and to the common urgency of action when threats have immediate, Internet-wide effects.

Further research is needed to address some challenges. For example, the diversity of academic disciplines involved likely will lead to different people using the same or similar terms in different senses or meanings. This phenomenon is not new. However, future work could attempt to adapt computer-assisted semantics tools and standards to this problem, such as ontology repair developed for OWL (Web Ontology Language) [49].

Conclusions

The intrusion kill chain is a useful model for CND, but models in cybersecurity can benefit from refining a more structured approach borrowed and adapted from other sciences. The mechanistic approach to modeling provides the template to be adapted to the growing information security field. The philosophy of information science has unique problems that must be worked through as we apply modeling techniques to existing information security work. We have begun that work by identifying some of the ramifications of applying the mechanistic approach to the intrusion kill chain model and the common language for computer security incidents. The mechanistic approach readily yielded refinements to both models.

There is still further work to be done in many respects, but this initial work provides some guideposts for future work in modeling security and applying mechanistic thinking via the philosophy of information science. Cybersecurity is now regarded by respected practitioners like Dan Geer as being outside the comprehension of any single expert [30], and so the community needs more structured communication between experts for the field to be able to continue to progress with the necessary speed and accuracy to continue to be effective. Using mechanistic models will provide a richer and clearer language for us to communicate and move the field forward.

Acknowledgement

The authors would like to thank our anonymous reviewers for their constructive feedback. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Department of Homeland Security or the United States Department of Defense.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

This material has been approved for public release and unlimited distribution.

CERT® is a registered mark of Carnegie Mellon University.

orcid.org/0000-0001-9356-219X (Spring)

Funding

This material is based upon work funded and supported by Department of Homeland Security under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense. Separately, Spring has the support of the UCL Overseas Research Scholarship and Graduate Research Scholarship.

References

[1]. Giere RN. How models are used to represent reality. *Philos Sci* 2004;**71**:742–52.

[2]. Joint Chiefs of Staff. Information operations. Technical Report JP 3-13. United States Armed Forces, 2014.

[3]. Hatleback E, Spring JM. Exploring a mechanistic approach to experimentation in computing. *Philos Technol* 2014;**27**:441–459.

[4]. Larman C. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*, 3rd edn. Upper Saddle River, NJ: Prentice Hall, 2004.

[5]. Simon HA. (1996). *The Sciences of the Artificial*. Cambridge, MA: MIT Press,.

[6]. Hutchins EM, Cloppert MJ, Amin RM. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Lockheed Martin, Bethesda, MD, 2011.

[7]. Illari PM, Williamson J. What is a mechanism? Thinking about mechanisms across the sciences. *Eur J Philos Sci* 2012;**2**:119–35.

[8]. Machamer P, Darden L, Craver CF. Thinking about mechanisms. *Philos Sci* 2000;**67**:1–25.

[9]. Bechtel W, Richardson RC. *Discovering Complexity: Decomposition and Localization as Strategies in Scientific Research*, 1st edn. Princeton, NJ: Princeton University Press.

[10]. Alberts C, Dorofee A, Killcrece G, *et al*. Defining incident management processes for CSIRTS: A work in progress. Technical Report CMU/SEI-2004-TR-015. Software Engineering Institute, Carnegie Mellon University, 2004.

[11]. Basin D, Capkun S, Schaller P, *et al*. Formal reasoning about physical properties of security protocols. *ACM Trans Inf Syst Sec (TISSEC)* 2011;**14**:16.
[12]. Liu P, Zang W, Yu M. Incentive-based modeling and inference of attacker intent, objectives, and strategies. *ACM Trans Inf Syst Sec* 2005;**8**:78–118.
[13]. Morgan RM, Bull PA. The philosophy, nature and practice of forensic sediment analysis. *Prog Phys Geogr* 2007;**31**:43–58.
[14]. Metcalf L, Spring JM. Blacklist ecosystem analysis: Spanning Jan 2012 to Jun 2014. In: *Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security*, pp. 13–22. Oxford, UK: ACM, 2015.
[15]. Craver CF.. *Explaining the Brain: Mechanisms and the Mosaic of Unity of Neuroscience*. Oxford: Oxford University Press, 2007.
[16]. Caltagirone S, Pendergast A, Betz C. The diamond model of intrusion analysis. Technical report. Center for Cyber Intelligence Analysis and Threat Research, 2013.
[17]. Bejtlich R. (2004). *The Tao of Network Security Monitoring: Beyond Intrusion Detection*. Boston, MA: Pearson Education.
[18]. Howard JD, Longstaff TA. A common language for computer security incidents. Technical report SAND98-8667. Sandia National Laboratories, 1998.
[19]. Glennan S. Mechanisms. In: Curd M and Psillos S (eds), *The Routledge Companion to Philosophy of Science*. New York: Routledge, 2013, 420–28.
[20]. Williamson J. (2015). *Evaluating Evidence in Medicine*. https://blogs.kent.ac.uk/jonw/projects/evaluating-evidence-in-medicine/.
[21]. Carroll L. (1893). *Sylvie and Bruno Concluded*. New York: MacMillan and Co.
[22]. Cox DR. Role of models in statistical analysis. *Stat Sci* 1990;**5**:169–74.
[23]. Illari P. Mechanistic explanation: Integrating the ontic and epistemic. *Erkenntnis* 2013;**78**:237–55.
[24]. Axelsson S. The base-rate fallacy and its implications for the difficulty of intrusion detection. In: *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pp. 1–7. ACM, 1999.
[25]. Anderson R. Why information security is hard: an economic perspective. In: *Computer Security Applications Conference*, pp. 358–65. New Orleans, LA: IEEE, 2001.
[26]. Spring JM. Modeling malicious domain name take-down dynamics: Why eCrime pays. In: *eCrime Researchers Summit (eCRS)*, pp. 1–9. San Francisco: IEEE, 2013.
[27]. Bogen J, Woodward J. Saving the phenomena. *Philos Rev* 1988;**97**:303–52.
[28]. Glymour CN. *The Mind's Arrows: Bayes Nets and Graphical Causal Models in Psychology*. Cambridge, MA: MIT Press, 2001.
[29]. Shimeall T, Spring J. *Introduction to Information Security: A Strategic-based Approach*. Waltham, MA: Syngress Publishing, 2013.
[30]. Geer D. Cybersecurity as realpolitik. In: *Black Hat USA 2014*, Las Vegas, Nevada: UBM, 2014.
[31]. Bell D, LaPadula L. Secure computer systems: Mathematical foundations. Technical report ESD-TR-73-278. MITRE Corp, Bedford, MA, 1973.
[32]. Chismon D, Ruks M. Threat intelligence: Collecting, analysing, evaluating. Technical report, MWR InfoSecurity, London, 2015.
[33]. Shirey R. Internet Security Glossary, Version 2. RFC 4949 (Informational), 2007.
[34]. Seacord RC. *Secure Coding in C and C++*. Upper Saddle Ridge, NJ: Pearson Education, 2005.
[35]. Lyon G. *Nmap Network Scanning: The Official Nmap Project Guide To Network Discovery and Security Scanning*. Nmap Project, 2011.
[36]. MITRE (2012). *Common Vulnerability Enumeration*. http://cve.mitre.org (13 October 2016 date last accessed).
[37]. MITRE (2014). *Common Weakness Enumeration: A Community-developed Dictionary of Software Weakness Types*. http://cwe.mitre.org (14 October 2016 date last accessed).
[38]. ISO/IEC. Open systems interconnection – basic reference model: The basic model. Technical report 7498-1:1994(E). International Organization for Standardization and International Electrotechnical Commission, 1996.
[39]. Alpcan T, Başar T. *Network Security: A Decision and Game-theoretic Approach*. Cambridge University Press, 2011.
[40]. Spring JM. Toward realistic modeling criteria of games in internet security. *J Cyber Sec & Info Systems* 2014;**2**:2–11.
[41]. Segura J. (2014). *The Proof is in the Cookie*. https://blog.malwarebytes.org/malvertising-2/2014/11/the-proof-is-in-the-cookie/.
[42]. Preimesberger C. Why 'malvertising' has become a pervasive security risk. *eWeek*, March 22, 2015.
[43]. Kadane JB. *Principles of Uncertainty*. CRC Press, 2011.
[44]. Spring JM, Kern S, Summers A. Global adversarial capability modeling. In: *IEEE eCrime Researchers Summit*, pp. 22–42, Barcelona: Anti-Phishing Working Group, 2015.
[45]. Rasmussen R, Aaron G. Phishing activity trends report: 4th quarter 2014. Technical report. Anti-Phishing Working Group, 2015.
[46]. Kanich C, Weaver N, McCoy D, *et al*. Show me the money: Characterizing spam-advertised revenue. In: *20th USENIX Security Symposium*. San Francisco, CA: USENIX, 2011.
[47]. Manadhata PK, Wing JM. (2012). *Attack Surface Measurement*. http://www.cs.cmu.edu/~pratyuspratyus/as.html (31 October 2016 date last accessed).
[48]. Forrest S, Perelson AS, Allen L, *et al*. Self-nonself discrimination in a computer. In: *IEEE Symposium on Security and Privacy*, p. 202, Oakland, CA: IEEE, 1994.
[49]. Kalyanpur A, Parsia B, Sirin E, *et al*. Repairing unsatisfiable concepts in owl ontologies. In Y. Sure and J. Domingue, editors, *The Semantic Web: Research and Applications*. *(European Semantic Web Conference)*, pp. 170–84, ESSI (European Semantic Systems Initiative): Budva, Montenegro, 2006.

Appendix

The mechanistic diagrams in this article were created using Tikz in latex. We provide the commands to create a sample of them here in the hope it will help others unfamiliar with any drawing method get started. There are many other tools that could be used to make these diagrams.

First, in the preamble of the document, we have:

```
\usepackage{tikz}
\usetikzlibrary{arrows}
\tikzset{>=triangle 60} %more visible arrows
\usepackage{pgfplots} % recommended
\pgfplotsset{compat=newest}
```

For Fig. 2, we write the following inside a float environment:

```
\begin{tikzpicture}
\tikzstyle{bigcirc}=[circle, minimum width=96pt, draw, inner sep=8pt]
\tikzstyle{smcirc}=[circle, minimum width=18pt, draw, inner sep=0pt]
\tikzstyle{medcirc}=[circle, minimum width=56pt, draw, inner sep=4pt]
5\node[bigcirc][label={below:$A_p$}] (Advers) at (0,0) {};
\node[bigcirc][label={below:$T_p$}] (Target) at (7,0) {};
\node[smcirc] (AdvWeap) at (.5,.5) {$E_e$};
\node[medcirc] (TargetPC) at (6.6,0) {$C_e$};
\node[smcirc] (RAT) at (6.3,-.4) {$R_e$};
10\draw[->] (Advers) to node [below] {$1_{ep}$} (Target);
\draw[->][bend left=80] (Advers) to node [below] {$2_e$} (AdvWeap);
\draw[open triangle 90->][bend left=52, dashed][] (AdvWeap) to node [above] {$3_p$} (Target);
\draw[->][bend right=75, dashed] (Target) to node [left] {$3.1_e$} (TargetPC);
```

```
\draw[->][bend left=45, dashed] (AdvWeap) to node [above] {$3_e$} (TargetPC);
15\draw[->][bend left=25] (AdvWeap) to node [above] {$4_e$} (TargetPC);
\draw[->][bend right=60] (AdvWeap) to node [below] {$5_e$} (RAT);
\draw[<->][bend left=15] (RAT) to node [below] {$6_{ep}$} (Advers);
\draw[->][bend left=10] (Advers) to node [above] {$7_{ep}$} (TargetPC);
\end{tikzpicture}
```

And for Fig. 4 we write the following inside a float environment:

```
1\begin{tikzpicture}
\tikzstyle{bigcirc}=[circle, minimum width=96pt, draw, inner sep=8pt]
\tikzstyle{smcirc}=[circle, minimum width=18pt, draw, inner sep=0pt]
\tikzstyle{medcirc}=[circle, minimum width=56pt, draw, inner sep=4pt]
\node[medcirc][label={left:$Browser$}] (user) at (0,0) {};
6\node[smcirc][label={below:$Cookie$}] (cookie) at (0,0) {$C$};
\node[medcirc][label={below:$Website$}] (website) at (3,4) {};
\node[medcirc][label={above:$Ad~network$}] (adnet) at (8,3) {};
\node[bigcirc][label={below:$advert$}] (advert) at (8,-1) {};
\node[medcirc][label={below:$javascript$}] (malcode) at (8,-1) {};
11\node[bigcirc][label={below:$URL$}] (url) at (3,-4) {};
\node[smcirc][label={below:$exploit$},text width=2cm] (exploit) at (3.3,-4) {$Angler~kit$};
\draw[->] (user) to node [left] {$1:fetches$} (website);
\draw[->][bend left=20] (website) to node [above] {$2:requests~ad$} (adnet);
\draw[->][bend left=10] (adnet) to node [right] {$3:selects$} (advert);
16\draw[->][bend left=10] (website) to node [left] {$4:fetches$} (advert);
\draw[->][bend right=80] (advert) to node [right] {$6:runs$} (malcode);
\draw[->][bend right=10] (advert) to node [above] {$5:pushes$} (cookie);
\draw[->][bend right=20] (malcode) to node [left] {$7:using~C,~calculates$} (url);
\draw[->][bend left=15] (malcode) to node [below] {$8:fetches$} (url);
21\draw[->][bend left=80] (url) to node [left] {$9:hosts$} (exploit);
\draw[->][bend left=30] (exploit) to node [left] {$10:compromises$} (user);
\end{tikzpicture}
```