A Data Driven Approach for the Science of Cyber Security: Challenges and Directions

Bhavani Thuraisingham, Murat Kantarcioglu,
Kevin Hamlen, Latifur Khan
University of Texas at Dallas
Richardson, TX, USA

Tim Finin, Anupam Joshi, Tim Oates
University of Maryland, Baltimore County
Baltimore, MD, USA

Elisa Bertino
Purdue University
West Lafayette, IN, USA

***Abstract*—**This paper describes a data driven approach to studying the science of cyber security (SoS). It argues that science is driven by data. It then describes issues and approaches towards the following three aspects: (i) Data Driven Science for Attack Detection and Mitigation, (ii) Foundations for Data Trustworthiness and Policy-based Sharing, and (iii) A Risk-based Approach to Security Metrics. We believe that the three aspects addressed in this paper will form the basis for studying the Science of Cyber Security.

I. INTRODUCTION

Data is at the heart of science. Scientific disciplines such as Physics, Chemistry and Biology have used data collected from experimentations to validate various theories over the past several centuries. More recently, scientific breakthroughs in multiple disciplines have been facilitated by tools that help researchers analyze massive datasets. For example, a recent editorial by *Science* argues that "**science is driven by data**." There is an increasing consensus that many of the future scientific advances will depend on how well researchers can share data to tackle important problems. While the Science of Security (SoS) is evolving and there are different visions about what "science of cyber-security" means, if what has happened in other scientific disciplines is any indication, we believe that the SoS will be increasingly data driven. We already see evidence of the applicability and effectiveness of a data driven approach. For example, recent advances in cyber-security include gathering data about potential attackers such as their techniques, incentives, and internal communication structures. Solid theoretical foundations for a SoS can be built by using such data to verify or refute theories about cyber-security.

One such application of a data-driven approach to security is the development of situation-aware intrusion detection systems that can handle advanced persistent threats. At present such attacks are generally detected post facto by forensic analysis. That analysis typically involves experts who piece together evidence from a variety of system sensors and logs that, interpreted in context, suggest an attack. What is needed is an analysis that can be automated and built on a formal grounding that includes semantically rich descriptions in ontologies, rule-based reasoning grounded on formal logic, generative graph grammars, and reasoning under uncertainty across distributed components.

While data is crucial for inducing, validating or refuting SoS formal theories, the data itself has to be accurate and trustworthy. Furthermore, organizations may not be willing to share their data due to potential liability and privacy concerns. For example, disclosing that an organization has been hacked may result in lawsuits. In addition, organizations that have more advanced cyber-security capabilities and tools may want to keep their data highly confidential. Therefore we need techniques to determine the accuracy of the data as well as enforce policy-based data sharing. While there has been considerable research in developing these techniques in recent years, foundational aspects of data trustworthiness and policy-based information sharing have yet to be fully investigated. Another challenge in establishing a science for cyber security lies in our ability to conduct repeatable experiments. While the repeatability of experiments is crucial for natural sciences and is gaining attention in many computing disciplines, it has yet to be examined for cyber security.

Finally, it is important to note that even if our secure systems are based on scientific principles, it is unlikely that these systems will be completely secure in the foreseeable future. Therefore the techniques for developing secure systems must take into consideration the risks that are involved. This not only involves cataloging the different risks that may occur and then analyzing each according to its potential impact and the likelihood of its occurrence, but also requires considering how risks may change over time. Such a risk-based approach can then be used to come up with the security metrics necessary to establishing a science for security. We need to create a comprehensive game theoretical risk assessment framework that models the interaction between the different factors such as the adversary and the system defender.

II. PROBLEM

A. *Challenges*

Cyberspace is increasing in complexity with heterogeneous components, such as different types of networks, diverse computing systems and multiple layers of software. Conflicts among adversaries are rapidly moving into cyberspace, and targeting our cyber-infrastructure. Providing cyber security solutions for such conflicts and defending against cyber-attacks in a complex landscape is

thus a major challenge. While considerable progress has been made over the past decade to secure the cyber space, many of the tools, technologies, and techniques are designed and developed on an ad-hoc basis without an investigation of their foundational principles. To address this significant limitation in the design and development of secure systems, we need to carry out cyber security experimentation to develop theories. We believe that data produced as a result of experimentation is a critical resource for SoS.

There are three major aspects that must be investigated. One is to represent and reason about the data used for cyber security experimentation to detect and mitigate attacks. The second is ensuring that the data collected is accurate and secure. Third is to develop an approach for gathering risk-based security metrics, including risks that may change over time as the situation in cyberspace evolves. Actions and decisions that the defenders may take to decrease certain risks may increase other risks. The defenders need to evaluate the short-term and longer-term consequences of their actions and decisions based on dynamic assessment of risks. We believe that such a risk management approach will enable us to come up with security metrics necessary for a scientific discipline. What is needed is a data driven SoS framework which emphasizes the effective, representation, management, sharing of accurate data securely to support cyber security experimentation that would result in foundational theories. The challenges that need to be addressed include the following:

- ***Data driven science for attack detection and mitigation:*** Foundations for representing, integrating, reasoning over and analyzing data for detecting and mitigating threats.
- ***Data trustworthiness and policy-based sharing:*** Foundations for data trustworthiness based on data provenance; carry out formal policy analysis for assured information sharing; and explore repeatable experiments
- ***A risk-based approach to security metrics:*** Comprehensive game theoretical risk assessment framework for gatherings security metrics.

B. Principles

The following principles have to be explored in designing a data driven SoS framework.

- The first principle, which we refer to as *Increase Trust by Limiting and Isolating Functionality* (abbreviated as *Isolation Principle*), emphasizes that "less is more" and calls for smaller components with well-defined interfaces. This principle is based on well-established security principles such as the Principle of Least Privilege.
- The second principle, which we refer to as *Adaptive Multiple Layers of Security* (abbreviated as *Independence Principle*) applies the Isolation Principle both statically and dynamically by dividing systems into several layers.
- The third principle, which we refer to as *Artificial and Natural Diversity* (abbreviated as *Diversity Principle*), emphasizes the need for diversity defenses that present attackers with unpredictable targets.
- The fourth principle, which we refer to as *Learning Systems* (abbreviated as *Learning Principle*), emphasizes the need for systems to dynamically learn from past activities, data and even from users.

III. TOWARDS DEVELOPING A DATA DRIVEN FRAMEWORK

A. Data Driven Science for Detecting and Mitigating Attacks

State-of-the-art intrusion detection and prevention systems (IDPSs) perform signature-based monitoring to identify malicious activities and generate alerts. Present systems share two key limitations: they are unable to identify attacks whose signatures are not known and they are point-based solutions geared to defend a single target. While such systems are good at defending against known attacks and identifying attacks seeking to bring down a system, they are useless against advanced persistent threats (APTs) and low and slow attack vectors. The latter are increasingly the preferred methods for nation state, criminal, and non-state actors.

We need to define and address a new science of security hard problem: how can we build intrusion detection systems that are situation-aware and so that it can be used for APTs and Low/Slow vectors. Today such attacks are generally detected post facto by forensic analysis by experts who piece together evidence from sensors and logs that, interpreted in context, suggest an attack. This is essentially an art. We need to automate the analysis by putting it on a formal grounding that includes semantically rich descriptions in ontologies, rule-based reasoning grounded in formal logic, generative graph grammars and reasoning under uncertainty across distributed components. While our focus will be on the formalisms and fundamental science, we need to include the elements needed to translate this science into practice by creating a prototype system. We need to explore how this makes cyber defense more adaptive by reasoning over attack vectors, attack targets and knowledge of the elements of the system.

We need to create the foundations of systems that dynamically analyze heterogeneous streams of information to extract facts that populate and maintain a semantically rich knowledge base (KB) with information about the resources being protected, the attacks they are experiencing and new vulnerabilities they might have. These facts will be used to deduce the context of the system, the possibility of attacks, and potential mitigations. The data and knowledge sources from which such facts can be extracted include textual sources (e.g., NVD, blog posts, chat rooms), hardware level sensors such as power draw, IDS systems such as Snort, host-based sensors, as well as information from peer systems. The KB will be built on Semantic Web languages (e.g., OWL) supporting both rule-based reasoning and graph-based analysis. A modern enterprise has a host of point systems that act independently from one another – an IDPS that scans network traffic at the gateway, firewalls regulating connections, application specific gateways doing application specific deep packet inspection, host-based monitors like tripwire, malware scanners, identity

management and authentication systems (sometimes with bio-metrics), etc.

In more sophisticated systems (SIEMs), alerts and warnings from individual components are aggregated and dashboarded in an operations center. Network security analysts monitor these, a process described as watchstanding, permitting highly-trained analysts to look at the disparate pieces of information and see if they "click together" into a pattern suggesting an attack. The analyst is aided by her background knowledge regarding the context of the system (e.g., the applications installed, the system's normal behavior pattern) and the external world (e.g., "intelligence" about what new attacks that exist in the wild, or the adversary's tradecraft). A similar forensic analysis process is done post facto when an attack is suspected.

As a representative of the best of the state-of-the-art in such approaches, consider the Cyber Kill Chain idea (Modeled on the DoD Find, Fix, Target, Track, Engage, Assess (F2T2EA) kill chain for targeting hostile forces.that tries to capture the offensive actions that an adversary is likely to take in attacking a system. It tasks the analyst to look at (log) data from various elements of the system, potentially across organizational boundaries, for events that fit this chain. The approach, however, is far from adequate as it is essentially looking for the "superman" in every analyst, and cannot be scaled given the level of human expertise and involvement needed. Our research will move from this state–of–the–practice to a context/situation aware system that largely automates this sophisticated (forensic) analysis done by the analyst on the sensed/observed data.

a) Context Representation and Sharing

Our context model requires information on components of the system, network and host data, attacks, their means and consequences, attackers and their campaigns, time and temporal relations, geo-spatial entities and relations, user identities and profiles, and mitigation actions taken. In the distributed and dynamic environments that we want to protect, no single component will have complete knowledge about its context. Knowledge sharing is an effective mechanism to help agents to build contextual knowledge, but requires that independently developed components and agents must share a common ontology and communication language.

In most prior work, contextual information has been represented as data structures or objects in the implementation language. Such representations lack expressiveness and extensibility, are difficult to share across implementations, and lack native mechanisms to define equivalences and mappings. Encoding the information in a meta-language like XML can help, but such interchange languages operate at a syntactic level and provide inadequate support for semantic representation and interoperability, both essential to knowledge sharing and context reasoning [Che04]. This means that XML representations of security-related information such as Stix, SCAP, IODEF and OVAL are inadequate for our purpose, though they provide strong, community- and standards-based points of departure.

One could use the semantic web language OWL, a W3C standard, to represent the security-related context data as an ontology. It provides richer sharing models than database schemas or XML to enable sharing of context data. Moreover, using semantic web languages makes available extensive existing ontologies and data for relevant domains for integration. There is considerable potential to develop OWL ontologies that can interoperate with and eventually form the semantic underpinnings for cyber security data, as shown in our prior work [Und03, Jos13].

We have developed a preliminary ontology to represent cybersecurity information which is available at [Cyb12]. Its key concepts are the attributes of the attack itself as well as the system attacked. We take a "victim-centered" view, and describe attacks on systems, their means, consequences, and targeted products. We further classify an attack as a *security attack* or a *privacy attack*. An attack has two primary properties: the way in which the attack was executed and its effects. In our ontology, we define an overall *AttackAttribute* class with subclasses *Means* that encapsulates the ways and methods used to perform an attack and *Consequences* that encapsulates the outcomes. The *Product* class encapsulates the information of the hardware, software and vendor of the system under attack. We need to leverage this work and enhance it to describe attack campaigns, potential mitigations, possible reconnaissance steps, and data exfiltration. We need to leverage existing efforts such as STIX, OVAL etc. and model them in our OWL ontologies. Very few of these systems model the hardware elements and power measurements of the system, so that will be a key focus of our extension efforts. We need to also extend and incorporate ontologies that let us describe actions, constraints and consequences.

b) Dectecting Attacks by Reasoning

One approach to detect attacks is for the analyst to define rules based on the system context and the incoming data. For example, we might predict an ongoing attack if the system has software installed that has been recently discussed in hacker forums as a potential attack vehicle and if the process corresponding to that software is running and behaving anomalously. As another example, an attack may be likely if a new device is added to a system, its device driver runs, consumes a lot of CPU, stats the password file, and the system sends data to machines with which it has not connected in the past. In most existing approaches, implementations program the logic of context reasoning directly into the behavior of the system, leading to rigid, difficult to maintain systems. Our approach allows the logic of context reasoning to be implemented separately from the behavior of the system. A combination of description logic and rule-based logical inference is a feasible and scalable approach to allow this decoupling [Che03].

There are several advantages to this approach. It helps separate the high-level reasoning logic from the low-level functional implementation, allowing developers to modify or replace context reasoning components. It allows many well-defined logic models of general concepts such as time

and space [Che04] to be directly used. Most importantly, it allows analysts to specify rules over the sensed knowledge and data describing potential attacks and mitigations. This permits forensic detection of attacks such as APTs in real-time that today depend on post facto analysis. When context interpretation and attack detection rules are explicitly represented, meta-reasoning techniques can be developed to detect and resolve inconsistencies. *Note that a reasoner can also suggest possible mitigations. If the attack was on a particular service, it could try to move that service to a different host (assuming that the clients internally would know how to rediscover it and retain state). Our approach permits us to deploy such "moving target" approaches* based on the context instead of blind randomization.

However, a key challenge arises: how can rule-based reasoning be done when the underlying facts that feed the rules that in turn determine context are streaming? Moreover, streaming data and facts are not available for post facto processing. Latency in detecting significant events related to security state that affect context is a key problem. For example, editing a firewall rule could create the opening for an attack vector that needs to be recognized and plugged before it can be exploited. Does this need for fast reasoning mean that we will need to be limited to RDFS models that will be using subsumption-based reasoning only? Or is it the case that we can go further, perhaps to description logic or OWL-QL? There is evidence that even first order reasoning can be done in near real-time for service management issues in pervasive systems. An additional approach to handle scale is to use rule context guards, where groups of rules are only enabled in and applicable to particular contexts.

Another issue is that the fact as inferred by the sensed data may not be the fact in terms of which a context rule is defined, but the two may be related. Consider a rule stating that if the system sees a very long string as a payload for an application, it should check if the application opens outbound connections to blacklisted hosts. While this involves a relatively simple subsumption-based deduction, in general there could be more complex relationships between terms in the rules and the streaming facts. Such streaming reasoning is, to the best of our knowledge, virtually unexplored in literature. The best analogy here is with streaming databases. Our facts are streaming and the context and attacks are defined by a set of "standing rules" which need to be triggered in response to the streaming facts. Just as traditional database engines can deal with streaming data but are neither efficient nor scalable in this context, a standard rule-based engine will be the same. One approach is to pre-compute an expanded set of rules. In cases where the ontologies are known and relatively static, this can be an efficient approach [Las02, Wal08].

We need to also explore the possibility of building a scalable reasoner on streaming database systems such as TelegraphCQ by enhancing them to match streaming facts with possible inferences over them. This can be used to add new facts to the stream that arise from other constraints. In prior work [Wal08], we created a reasoner over streaming data for a subclass of OWL. Our results indicate that the stream processor-RDF reasoner combined is about three orders of magnitude faster than the traditional RDF reasoner. Moreover, while the time and memory needed by traditional reasoners increases as new facts stream in, our processing time stays constant and significantly improves memory performance. However, this approach is restricted for now to subsumptions. We need to create a scalable stream reasoner to speed up the rule evaluation process, especially exploring the expressivity and speed tradeoffs as we move from RDF to various OWL profiles.

c) Detecting Attacks using Graph Grammars

Another natural way to represent activity involving cyber infrastructure is with a dynamic graph entailed by its OWL representation, with nodes representing entities (programs, ports, routers, hosts and users) and edges modeling events and interactions (running a program, opening a port, an attacker initiating a campaign). As new interactions involving known or new resources occur, or new information becomes available about entities and interactions, the graph is updated in real-time. We need to leverage our prior work on graph grammars to develop powerful new methods for representing, reasoning about, and detecting attacks. In contrast to the more familiar string grammars that define sets of strings or probability distributions over strings, graph grammars define sets of graphs or probability distributions over graphs. Graph grammar production rules have sub-graphs on their right-hand sides that are used to rewrite non-terminal nodes or edges during the derivation of a graph. Note that this rewriting process naturally captures two key aspects of the domain - context and dynamics. Production rules encode contextual information in their right-hand sides by embedding non-terminals in a sub-graph of appropriate scope; and the application of production rules elaborates the structure of the graph over time.

We have developed efficient methods for learning graph grammars (both structure and parameters) from data. Given a collection of graphs corresponding to network flows in an attack, it is possible to learn a grammar that represents the distribution over graphs from which the training data were sampled. The grammar can then be used to (1) parse new graphs and determine the probability that they were drawn from the same distribution (i.e., recognize new attacks), (2) identify structure that dramatically increase the probability of the graph if it were present (partial parsing), thus providing an "early warning" and focusing information collection efforts, and (3) expose compositional structure in the domain for human consumption as production rules extracted from data. Expert domain knowledge is also easily encoded as hand-written production rules that can be refined by the learning algorithms. This is in contrast to the analysis of attack graphs to infer patterns defined in literature, which is limited to edit distances between instances.

Existing algorithms will need to be extended in a number of ways to be effective in the target domain. One of the central operations of the learning algorithm is finding common subgraphs. The current implementation performs an exact match, but ontologies defined over entities and

relations can allow for more effective and efficient use of the data by matching on common generalizations. For example, rather than treating ports 22 and 23 as different for matching purposes, the ontology may unify them as ports used for remote login and allow subgraphs containing them that are otherwise the same to be matched. Existing work on graph grammars focuses on context-free productions, in which a non-terminal can be rewritten using any rule with a matching left-hand side. Our prior work on learning mildly-context sensitive string languages suggests that in some cases there are a few important but computationally tractable context-sensitivities. We need to identify the central contexts in cyber infrastructure attacks and seek ways of making them tractable for graph grammars as well. Finally, existing learning algorithms allow for static labels on nodes and edges, but not for probability distributions over labels. It would be useful, for example, to learn and represent a normal distribution for the hourly number of packets that flow along a communication link, or a multinomial over the ports that are used by an application.

d) Detecting Attacks Using Stream-based Classification

We treat attack detection as a *data stream* classification problem. Data streams have a dynamic nature, which brings about the problems of *concept-drift* and *concept-evolution* [Mas13]. *Concept-drift* occurs as a result of a change in the underlying concept of the data (e.g., evolution of attacker methods, or evolution of benign software usage patterns). It makes previously trained models outdated, and therefore necessitates the continuous refinement of the classification model in response to new incoming data. Concept-evolution, on the other hand, refers to the emergence of novel classes over time. For example, a new APT with new attack methodology and objectives would typically fall into this category.

In our approach, we assume that the data stream (e.g., sensor information) is divided into equal-sized chunks. The heart of the system will be an ensemble of classifiers. When a new unlabeled test instance arrives (part of a chunk), the ensemble will be used to classify the instance. If the test instance is identified as an outlier, it will be temporarily stored in a buffer for further inspection. Otherwise, if it is not an outlier, then it will be classified with a known label. The buffer will be periodically checked to see whether a novel class has appeared. If a novel class is detected, the instances belonging to the novel class will be identified and tagged accordingly. We need to also apply collective classification over multiple data sources based on Markov network to facilitate inference. The outputs will be another type of input to the stream based reasoning tools.

e) Text Analysis to Detect Emerging Cyber Threats

The Web is often our first source of information about new software vulnerabilities exploits and cyber-attacks. In addition to curated sources such as NVD, there are informal sources such as hacker blogs and forums, chat rooms and social media. Even though these are noisy and redundant and contain misinformation, they can be mined and aggregated to provide early warnings of new vulnerabilities and attacks, track the evolution of existing ones, produce evidence for attribution and estimate the prevalence and geographical distribution of known problems. We have developed preliminary systems [Mul11, Jos13] demonstrating the feasibility of automatically generating RDF-linked data from vulnerability descriptions in text sources. The current prototype uses a CRF-based system to extract information on cybersecurity-related entities, concepts and relations that is then represented using custom ontologies for the cybersecurity domain and mapped to objects in the DBpedia knowledge base using DBpedia Spotlight and the Wikitology knowledge base [Fin10]. Our evaluation shows the approach to be promising [Jos13].

We need to broaden applicability, improve accuracy and create a sustainable system that can deal with evolving and emerging threats. For example, we need to create a corpus of relevant text that can be used to develop and train language models for the cyber security domain. We also need to also create a framework to dynamically collect text from news streams (e.g., blogs) that we can analyze to detect potential new vulnerabilities and threats. Our annotation framework and crowd-sourcing approach [Fin10a] has to be leveraged to work with this corpus and use it to improve the accuracy of our entity and relation extraction system for cybersecurity text using probabilistic approaches [Sle13]. A system that can process new text from our ingest system on a continuous basis and produce a stream of linked data assertions to update our knowledge base has to be developed.

f) Handling Inconsistency

When different security components are built to share knowledge and acquire context from sensed data, we face the problem of receiving noisy and often inconsistent information. We need to address the many problems involving representing and using both data and knowledge that that is uncertain. Uncertainty will be introduced at almost every turn in our system. The underlying sensors are themselves imprecise and depend on empirically determined thresholds and so may provide inconsistent information. Distributed host and network sensors can reach incompatible interpretations of a situation due to contextual differences. For example, a host that has seen a flood of spurious DNS responses might be sharing information with a peer that is (yet) unattacked. Information extracted from text may be wrong due to errors in the linguistic analysis or because the text itself has errors in it due to being out of date or from an unreliable source. Inference rules may themselves be heuristics that are accurate most, but not all, of the time. Many of our graph-based analytics and machine-learning components will produce results with associated certainty measures or probabilities. We need to draw on a range of general approaches to managing such inconsistent data,, including assumption-based abductive reasoning, modeling data provenance and reasoning about reputation and trust, machine learning for outlier detection, argumentation protocols for resolving inconsistencies, and custom query probes to find inconsistent data. Scalable approaches such as those based on Markov Logic Networks

and Bayesian Logic are needed to handle large volume and velocity data.

B. *Foundations for Trustworthy Data*

Our objective is to explore the foundations of data trustworthiness as well as assured information sharing so that accurate and trustworthy data is provided to carry out cyber security experiments. In addition, we need to introduce a novel idea of the repeatability of cyber security experiments.

Data Trustworthiness: It is critical to provide comprehensive solutions for assessing and assuring the trustworthiness of the collected information to carry out cyber security experiments (as well as for several other applications). Attacks may result in inaccurate data being provided to decision makers and analysts. Our goal is to address this problem by developing a framework for assessing and assuring the data trustworthiness.

Assured Information Sharing: Daniel Wolfe (formerly of the NSA) defined assured information sharing (AIS) as a framework that "provides the ability to dynamically and securely share information at multiple classification levels among U.S., allied and coalition forces." Furthermore, Lonny Anderson, the Chief Information Officer of the NSA, has stated that the agency is focusing on a "cloud-centric" approach to information sharing with other agencies [Nsa11] and has recently stated that "the leaks actually reinforced the need to move to the cloud and move there quickly" [Sef13]. Development of a framework to securely share the cyber security data is being pursued by NIST under a Presidential directive. While various assured information sharing strategies and systems have been developed during the past decade, the foundations for assured information sharing is yet to be established.

Repeatability of Experiments: As in natural sciences, it is critical that experiments be described in detail and data used and generated by experiments be made available so that experiments can be repeated and generalized. Reproducibility is crucial to validate results, discover errors and detect scientific frauds. A similar principle is emerging in the Computer Sciences. As discussed by Freire et al. [Fre12], "*in a computational environment, it should be possible to repeat a computational experiment as the authors have run it or change the experiment to see how robust the authors' conclusions are to changes in parameters or data (a concept called workability).*" While research on reproducibility is active in certain areas of Computer Science, it has yet to be articulated in cyber security. We are investigating this problem.

a) Trustworthiness of Data Based on Provenance

To correctly identify which data can or cannot be trusted, we are investigating a novel method, based on the iterative filtering technique [Ker10] and provenance. Provenance can be physical (e.g., geographical locations, network nodes in Internet) or logical (e.g., business entities in workflow systems, roles in a social network). Provenance is important in that data provided by a trusted source and manipulated by trusted system components can be deemed trustworthy. There is thus inter-dependency between data, and data sources and system components with respect to the assessment of their trustworthiness, i.e., trustworthiness of the data affects the trustworthiness of its source and components that manipulated the data, and vice-versa. We have developed a preliminary approach to determine the trustworthiness of data in systems based on provenance. It includes the computation of confidence levels and similarity measures. For example, the trustworthiness level of a component is determined by confidence level of its related data items. Furthermore, we also compute provenance similarity. Details are given in [Thu16].

Accurate evaluation of trustworthiness: The key component to achieve high accuracy is the similarity measure technique. In addition to our preliminary approach which uses the normal distribution, we need to develop a more elaborate probability model to evaluate data similarity. We also need to apply various graph topology matching techniques to improve the accuracy of provenance similarity.

Protection from collusion: Current approaches based on iterative filtering techniques are not able to protect against collusion. We need to investigate how to extend our initial approach by using more robust variance estimation methods.

Efficient processing for very large-scale systems: Evolving trustworthiness in the cyclic framework can be done whenever a new data item arrives. We call this approach the *immediate mode*. The immediate mode immediately reflects change in information; however it is not applicable for large-scale systems. To achieve reasonable performance, we need to develop a *batch mode* which periodically updates the trustworthiness levels only once for a considerable number of accumulated data items, while providing comparable accuracy to the immediate mode.

b) Formal Policy Analysis

We have developed multiple assured information sharing systems under the common MURI project. One such system shares data in the cloud. In this approach, the data and policies are represented in RDF and stored in the cloud. We developed a SPARQL query engine to query the data and a RDF policy engine to process the information sharing policies [Cad12]. One limitation of our approach is the lack of a formal policy analysis for information sharing. Our current work focuses on applying formal policy analysis for privacy protection in federated environments. We believe that such formal policy analysis [Ham12a] will be needed for addressing limitations in assured information sharing.

Our information sharing system is applicable to a variety of mission-critical, high-assurance data sharing problems that span multiple, mutually-distrusting organizations, data sources, and principals. In order to provide maximal security assurance in such settings, it is important to establish strong formal guarantees regarding the correctness of the system and the information sharing policies it enforces. To that end, we are developing an infrastructure for constructing formal, machine-checkable proofs of important system properties and policy analyses for our system. Machine-checkable proofs constitute the highest level of rigor for verifying security properties of software systems because, unlike human-readable proofs, they can be

automatically validated by proof-checking software that has undergone extensive manual verification by the mathematical community. Thus, such proofs reduce the trusted computing base of the systems they concern down to the foundations of mathematics [Kau04]. While machine-checkable proofs can be very difficult and time-consuming to construct for many large software systems, our choice of SPARQL, RDF and as query and policy languages, respectively, opens unique opportunities to elegantly formulate such proofs in a logic programming environment. Our prior work on machine-checking program-rewriting software security systems in Prolog (e.g., [Sri11]) has resulted in a powerful model-checking and formal reasoning system that we are applying to the analogous query-rewriting validation problem posed here. This system leverages recent advances in co-logic programming [Gup07] to encode co-inductive reasoning that is otherwise difficult to express in standard programming environments. We therefore encode policies, policy-rewriting algorithms, and security properties as a rule-based, logical derivation system in Prolog, and to apply these model-checking and theorem-proving systems to produce machine-checkable proofs that these properties are obeyed by the system.

Policy-rewriting in our context can be understood as a special case of in-lined reference monitoring [Sch00]. In-lined Reference Monitors (IRMs) enforce software security policies by in-lining the logic of a security monitor (e.g., for enforcing access control) directly into untrusted code. This results in self-monitoring code that can safely be executed on systems that lack native controls adequate to enforce the desired policy. To support data sources that do not trust the cloud-based information sharing system, we plan to supplement our system with a certifying IRM framework. Certifying IRMs allow code or query recipients to automatically and independently verify the policy-compliance of code or queries of untrusted provenance [Ham06, Sri11]. In the context of our information sharing system, this introduces at least two major opportunities for trusted computing base reduction:

- It allows data sources to receive mediation from untrusted mediators by independently certifying the local policy-compliance of the requests they receive.
- Policy enforcement (or re-enforcement as a second line of defense) can be pushed down to the level of the data management system.

Our certifying IRM-based protections will support bytecode-based data management systems such as Hadoop MapReduce [Dea08]. In prior work we have developed extensive certifying IRMs for enforcing data access controls in similar environments, including those based on Java [Ham12] and Microsoft .NET [Ham06], and we have implemented supplemental protection systems for Hadoop in the past [Kha10]. This makes us well-equipped to conduct the research. The machine verification strategy discussed in the previous paragraph leads directly to the development of the certifying IRM because it establishes the correctness of the certification procedure; in many cases the verifier's implementation follows directly from the proof. Our prior work has developed machine certifiers for IRMs based on type-checking [Ham06] and model-checking [Ham12].

c) Reproducibility of Security Experiments

To tackle the problem of reproducibility of security experiments, we need to analyze scientific literature in the area of security to determine which specific security topics involve experimental analysis and the types of experiment that are carried out. We need to also determine commonly used datasets by the security research community. We need to then extend available reproducibility tools or develop new tools for security experiments and apply these tools to specific security attacks and defenses, such as return-on-programming attacks and the corresponding defense techniques (e.g. randomization) and intrusions and the corresponding intrusion detection techniques.

The development of a methodology and tools for the reproducibility of security experiments involves addressing many issues, including:

Models for representing security attacks: As security attacks are often carried out through several steps, one has to describe not only each single step but also the step sequence and the time interval between each step. Attack graphs could be an initial model that could be extended by including all relevant information needed for each attack step. In addition, as in many cases the attacks are against a specific software system (as for example, a SQL injection attack against an application), the execution of the attacked system or application should be also reproduced. As in many cases, this may be difficult, the issue of fidelity of the reproduced attack is critical in that we need to understand what details concerning an experiment need to be preserved and which ones can be discarded.

Human models: As security attack and defense experiments may involve humans, the challenge is how to reproduce experiments involving humans, as it may be difficult or not possible to involve the same humans involved in the initial experiment. It is thus relevant to investigate how to represent human behavior through software agents.

Composability of Experiments: As a system may undergo different attacks at the same time, it is critical to perform security experiments in which multiple pre-existing experiments are combined. This is crucial in order to analyze different attacks and defenses in combination. For example, the use of a given security defense may negatively impact other security defenses. We believe that these types of experiments may result in important insights (e.g., the best defenses to protect against multiple attacks).

Management of experiment repositories and visualization of experiments: The challenge is to develop query languages able to retrieve experiments and compare experiments. Also visualization of experiment steps is very important especially for educational activities.

C. *Security Metrics: A Risk Analysis-based Approach*

One of the main challenges that arises in establishing a science for cyber-security is to develop metrics to measure the security of the systems. Unfortunately, coming up with one "number" that measures security is problematic since security of a system will depend on who attacks the system,

the resources used by the attacker, system internals, human factors, etc. Due to the dynamic nature of the attackers, any risk assessment based on the statistics learned from past data can quickly become obsolete. For example, intrusion data collected in the past may not contain any information related to new zero day attacks. On the other hand, any defensive mechanism employed by the defender may cause attackers to adapt and change their strategy. For these reasons, we need to model the entire interaction between the attackers and defenders as a game. Basically, an attacker or group of attackers will be represented by their potential objective and/or utility, technical capabilities (e.g., which resources that they can launch an attack) and resources (e.g., the number of simultaneous attacks that they can launch). We have pioneered techniques for incorporating risk into botnet detection and assured information sharing [Ben10]. We have modeled attacks using differential games theory [Kan11], [Hoe12]. Our work is rather unique in that we not only consider adversary behavior models, but also the epidemic issues presented in the cyber security domain through network connections. We further invoke mean field game theories to study our unified risk framework.

a) Modeling Attackers and Their Capabilities

The first step in our risk assessment-based security measurement approach will be to model the attacker. Clearly, different types of attackers will have varying goals. For example, certain state actors may be more interested in gathering sensitive information covertly. On the other hand, different non-state attackers may want to maximize damage. We plan to capture these varying goals in terms of different utility functions. While creating these utility functions, we need to leverage the existing work in behavioral game theory related to behavioral heuristics such as loss aversion [Cam03]. Another important aspect of attacker modeling is the capture and representation of his/her capabilities. For example, aspects to consider in this activity include modeling the different capabilities (e.g., attacking machines), modeling the different utilities and behaviors**,** **e**xploring ways to determine a strategy's effectiveness and its applicability constraints such as computational requirements and measuring the cost of the strategy**.**

Since it is hard to know the precise technical capabilities of an attacker in advance (e.g., the number of unknown zero day attacks), we plan to represent attacker's capabilities as possible systems and/or components that can be attacked successfully with certain probability. In addition, we need to also model the potential effect of the successful attack. For example, for certain state actors, we can represent their capabilities as a risk profile by estimating certain attack related parameters. Examples of such parameters we plan to consider include the probability that the attackers have zero day vulnerability based attacks that can be used against certain computer systems (e.g., Windows 8), the probability that attack can be exploited remotely and the likelihood of detecting attacks based on predictive analytics as well as replicating the attacks the cost of replication.

Other parameters need to be estimated to reason about the potential impact of an unknown attack. By changing and stress testing these parameters, we need to understand the capabilities of a wide range of attackers with varying goals.

b) User Risk Modeling

There are different types of users including malicious users and normal users. These malicious users cause the insider threat problem and will be considered to be attackers. In addition, due to the normal user errors (e.g., successful phishing attacks), system resources and/or nodes could become vulnerable. We need to handle these problems by considering potential user involvement in an attack either as a facilitator and/or a direct perpetrator. To address the risks due to a malicious user, we need to model the malicious user and/or insider as an attacker who has already gained control of a system. This will enable us to use our attacker models with slight modifications. In addition, we need to consider human/user errors as another attack vector. For example, a computer that is running Windows 7 could become compromised due to a user who clicks on a malicious link. At the same time, compared to the generic zero day exploit, such user error based attack may not be replicated as easily. Therefore, we plan to address user error based risks as a type of an exploit/attack that may be harder to replicate. These attacks and errors will cause potential risks to the normal users when the underlying system is compromised. Basically, such compromised systems not only endanger the success of the mission but also the safety of the normal users. We need to consider such situations in our holistic approach to risk assessment.

c) Modeling Network Risks under Attacks

In order to better estimate the risks associated with certain attacks, we need to understand how such attacks could affect the computer networks to be protected. For example, we need to understand the impact of a zero day attack affecting machines running certain types of software attached to the network. To achieve this goal, we model computer networks by leveraging ideas from social network analysis. Basically, over the years, many probabilistic models have been developed to model how ideas, diseases, etc. spread over the networks. For example, models have been developed to classify social network data using a combination of node details and connecting links in the social graph [Sen08]. We plan to use similar ideas to estimate the impact of an attack on a subset of computer and/or systems on a given computer network. Our model will be composed of three risk assessment models: a local risk assessment model, a relational risk assessment model, and a collective inference model.

Local risk assessment models The local risk assessment model will probabilistically examine details of a computer and/or node and constructs risk estimation based on the details of that node. For instance, we can have a simple probabilistic model and/or probabilistic rules that specify the impact of a zero day attack on a particular system based on the attributes of this network node. In addition, these local models will consider interactions between the different system components of the node. This is important in order to assess the vulnerability of a specific node, since the different components could be exploited to launch a

successful attack. We need to explore different probabilistic models (e.g., simple Bayesian belief networks) that are suitable for building effective local risk assessment models. In addition, we need to explore the best way to integrate the results of the detection strategies and/or alerts into the local risk assessment models.

Relational risk assessment models The relational risk assessment model is a separate type of probabilistic model that looks at the neighborhood structure of a given network node, and uses the local risk assessment output for neighboring nodes to estimate how the neighbors of a node can affect its security risk. To build this relational risk assessment model, we plan to leverage tools such as Markov logic networks where we can also incorporate domain heuristic/rules for estimation [Ric06]. For instance, using past data or domain expert, we can infer that if a node N_i is attacked then the probability that node N_j can still work correctly is p_{ij}. In addition, we need tools to determine dependencies among the different parts of the network could be learned from data. Here, we would like to emphasize that learning such probabilistic rules based on the past data is used to understand the dependencies in the existing network and infrastructure and it is not directly related to ever changing attack data and/or information. In addition, we need to explore how the attribute similarity (e.g., software and/or hardware components used across nodes) can be incorporated into our models.

Collective Inference The previous tools discussed are not sufficient by themselves to understand and assess all the dynamic risks under a novel attack scenario. Local risk models consider only the details of the network node under a given attack scenario. Conversely, relational models consider only the link and neighbor structure of a specific node. Collective inference-based models attempt to make up for these deficiencies by using both local and relational models in a precise manner to attempt increasing the risk assessment accuracy in the network. We need to explore how to combine such risk estimations with mean field theory based approaches to make it faster in practice. Mean Field Theory is an averaging technique, coming from physics, which results in simplifications [LAS07].

d) Modeling Defensive Strategies

Similar to attack models, we need to model defensive strategies based on their cost, effectiveness and applicability to specific attack types. We need to explore multiple classes of defensive strategies to create profiles that can represent their properties. While creating these defensive option profiles, we need to specially focus on important aspects related to network. Due to the diverse properties of the different networks and systems, it is important to understand the computational and/or power needs of the different defensive strategies. For example, certain defensive strategies (e.g., randomizing binaries) may not be applicable and/or costly (e.g., requires too much power consumption) for certain nodes. In the following, we provide a simplified mathematical model preview. In this version, concentrating on analyzing the defender's optimal defense strategy in anticipation of various classes of attack, we consider the network system to be protected as one entity, and fix the attack severity. This entity corresponds to the command center of the network system. The command center defines a defensive strategy to best protect the assets in anticipation of severity of attacks, i.e., different classes of attack.

Solving this problem provides a nominal strategy which can be used as a preliminary starting point. To address the ever-changing attack parameters, defensive strategies will be defined for each node of the network as a function of the available information.

e) Combining All Parts: Game Theoretic Models for Holistic Risk Assessment

All the previous components need to be combined in an extensive game. The attacker tries to optimize his attack by taking into account the defender's existing defensive strategies, and network information. This game theoretical model will be able to answer questions by approximately analyzing the equilibrium behavior under different scenarios obtained by varying parameters in our models. For example, the impact of different zero day attacks could be estimated by analyzing the change in equilibrium performance. In addition, different defensive strategies will be tested to estimate their effectiveness. Of course, doing these estimations efficiently will require novel approaches in combining simulation-based schemes with analytical approaches such as mean field theory.

Mean field theory [Las07] is a novel approach to take into account the influence of large communities on the behavior and the decision of a single individual. In our context, the safety of a single system depends on the interaction level of the global network it is connected to. The attacker will have to penetrate the defenses to reach a precise target. This connection to the community cuts both ways, but it is not neutral. When the network is very large, one can approximate the interaction by an averaging effect, the mean field term. This simplifies greatly the analytic treatment. We would like to investigate in this research the possibilities offered by Mean field theory in cyber security, which has not been considered extensively before. Our approach is detailed in [Thu16].

IV. SUMMARY AND DIRECTIONS

As stated earlier, while the Science of Security (SoS) is evolving and there are different visions about what "science of cyber-security" means, if what has happened in other scientific disciplines is any indication, we believe that the SoS will be increasingly data driven. We see evidence of the applicability and effectiveness of a data driven approach. For example, recent advances in cyber-security include gathering data about potential attackers such as their techniques, incentives, internal communication structures. This paper describes solutions and directions for a data driven framework to studying the science of cyber security. In particular, it focusses on: (i) Data Driven Science for Attack Detection and Mitigation, (ii) Foundations for Data Trustworthiness and Policy-based Sharing, and (iii) A Risk-based Approach to Security Metrics.

We believe that what we have discussed here is the first step towards a data driven approach to SoS. There are areas in cyber security that will benefit from such an approach including the study of the inference problem, data privacy, and insider threat detection. As we make progress towards developing solutions for a data driven approach for SoS, we believe that additional challenges will be uncovered. As stated earlier, data is at the heart of science and to study the science of cyber security data will be a crucial aspect.

ACKNOWLEDGMENT

We acknowledge the support of AFOSR (contract MURI FA9550-08-1-0265) for this work. We thank Dr. Robert Herklotz for his encouragement.

REFERENCES

[Ben10] A Bensoussan, M Kantarcioglu, S Hoe, A Game-Theoretical Approach for Finding Optimal Strategies in a Botnet Defense Model, GameSec 2010.

[Cad12] T Cadenhead, V Khadilkar, M Kantarcioglu, B Thuraisingham, A cloud-based RDF policy engine for assured information sharing". SACMAT 2012.

[Cam03] C Camerer, Behavioral Game Theory: Experiments in Strategic Interaction", Princeton University Press, 2003.

[Car12] R Carmona, F. Delarue, A Lachapelle, Control of McKean-Vlasov Dynamics versus Mean Field Games, Working paper, 2012

[Che03] H. Chen et al. An ontology for context-aware pervasive computing environments. The Knowledge Engineering Review 18, 2003.

[Che04] H. Chen. An Intelligent Broker Architecture for Pervasive Context-Aware Systems. PhD thesis, Univ. of Maryland, Baltimore County, Dec. 2004.

[Cyb12] Cybersecurity IDS ontology. http://ebiquity.umbc.edu/ontologies/cybersecurity/ids/.

[Dai08] C. Dai, D. Lin, E. Bertino, and M. Kantarcioglu, An Approach to Evaluate Data Trustworthiness Based on Data Provenance, Proc. of the 5th VLDB Workshop on Secure Data Management, Auckland, New Zealand, Aug. 2008.

[Dea08] J. Dean and S. Ghemawat, MapReduce: Simplified data processing on large clusters, Communications of the ACM, 2008.

[Fin10] T. Finin and Z. Syed. Creating and Exploiting a Web of Semantic Data. 2nd Int. Conf. on Agents and Artificial Intelligence. Springer, Jan. 2010.

[Fin10a] T. Finin et al. Annotating named entities in twitter data with crowdsourcing. Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, NAACL HLT, 2010.

[Fre12] J. Freire, P. Bonnet, and D. Shasha, "Computational Reproducibility: State-of-the-Art, Challenges, and Database Research Opportunities," Sigmod'12, May 20-24, 2012, AZ.

[Gup07] G. Gupta et al, Coinductive Logic Programming and Its Applications.In Proceedings Int. Conf on Logic Programming, 2007.

[Ham06][K. Hamlen, G. Morrisett, and F. Schneider, Certified in-lined reference monitoring on.NET. rs , Proceedings ACM SIGPLAN Workshop, 2006

[Ham12] K. Hamlen, M. Jones, and M. Sridhar, "Aspect-oriented runtime monitor certification. Proceedings of TACAS, 2012.

[Ham12a] K Hamlen, L Kagal, M Kantarcioglu, Policy Enforcement Framework for Cloud Data Management, IEEE Data Eng. Bull. (DEBU) 35, 2012.

[Hoe12] S Hoe, M Kantarcioglu, A Bensoussan, A Game Theoretical Analysis of Lemonizing Cybercriminal Black Markets, GameSec 2012.

[Jos13] A. Joshi at al. Extracting cyber-security related linked data from text. 7th IEEE Int. Conf. on Semantic Computing. IEEE Computer Society, Sept. 2013.

[Kan11] M. Kantarcioglu, A. Bensoussan and S. Hoe, "Impact of Security Risks on Cloud Computing Adoption", Alerton Conference on Communication, Control, and Computing, 2011

[Kau04] M Kaufmann and J. Moore, Some key research problems in automated theorem proving for hardware and software verification, Revista de la Real Academia de Ciencias, 2004.

[Ker10] C. de Kerchove and P. Van Doreen, "Iterative Filtering in Reputation Systems", SIAMJ. Matrix Anal. Appl., Vol.31, No.4, March 2010

[Kha10] A Khaled, M. Husain, L Khan, K. Hamlen, and B Thuraisingham, A token-based access control system for RDF data in the clouds, In Proceedings of IEEE Cloudcom, 2010.

[Las02] O. Lassila. Taking the RDF model theory out for a spin. Int. Semantic Web Conf. 2002.

[Las07] J.M. Lasry and P.L. Lions, "Mean field games", Jpn. J. Math., vol. 2(1), 2007.

[Lel08] M. Lelarge and J. Bolot, A Local Mean Field Analysis of Security Investments in Networks, Proceedings Third International Workshop on Economics of Networked Systems, 2008.

[Mas13] M. Masud, L. Khan, J. Han, et al. Classification and Adaptive Novel Class Detection of Feature-Evolving Data Streams. IEEE Trans. Knowl. Data Eng. 25, 2013.

[Mul11] V. Mulwad et al. "Extracting Information about Security Vulnerabilities from Web Text" Web Intelligence for Information Security Workshop. IEEE 2011.

[Nsa11]http://www.informationweek.com/news/government/cloud-saas/229401646

[Ric06] M Richardson, P. Domingos, Markov Logic Networks, Machine Learning, V62, 2006.

[Sch00] F. Schneider, Enforceable security policies, ACM Transactions on Information and System Security, 2000.

[Sef13] G. Seffers, Committed to Cloud Computing, SIGNAL Magazine, October 2013.

[Sen08] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. AI Magazine, Vol. 29, 2008.

[Sle13] J. Sleeman and T. Finin. Type Prediction for Efficient Coreference Resolution in Heterogeneous Semantic Graphs. 7th IEEE Int. Conf. on Semantic Computing. Sept. 2013.

[Sri11] M Sridhar and K Hamlen, Flexible in-lined reference monitor certification: Challenges and future directions Proceedings ACM SIGPLAN Workshop on Programming Languages meets Program Verification (PLPV), 2011.

[Thu16] B. Thuraisingham et al, A Data Driven Approach for the Science of Cyber Security: Challenges and Directions, Technical Report, the University of Texas at Dallas, June 2016.

[Und03] J. Undercoffer et al. Modeling Computer Attacks: An Ontology for Intrusion Detection. 6th Int. Symposium on Recent Advances in Intrusion Detection. Springer, September 2003.

[Wal08] O. Walavalkar et al. Streaming Knowledge Bases. 4th Int. Workshop on Scalable Semantic Web Knowledge Base Systems, Oct. 2008.