2013 IEEE Security and Privacy Workshops

Bridging the Gap: A Pragmatic Approach to Generating Insider Threat Data

Joshua Glasser
ExactData, LLC
Rochester, NY
joshua.glasser@exactdata.net

Brian Lindauer
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA
lindauer@sei.cmu.edu

***Abstract*—The threat of malicious insider activity continues to be of paramount concern in both the public and private sectors. Though there is great interest in advancing the state of the art in predicting and stopping these threats, the difficulty of obtaining suitable data for research, development, and testing remains a significant hinderance. We outline the use of synthetic data to enable progress in one research program, while discussing the benefits and limitations of synthetic insider threat data, the meaning of realism in this context, as well as future research directions.**

I. Introduction

Malicious insiders are current or former employees or trusted partners of an organization who abuse their authorized access to an organization's networks, systems, and/or data. Insider threats, the malicious acts carried out by these trusted insiders include, but not limited to, theft of intellectual property or national security information, fraud, and sabotage. Many government, academic, and industry groups seek to discover and develop solutions to detect and protect against these insider threats.

A significant impediment to these programs is the lack of data to analyze. To be useful, this data must contain a detailed account of human behavior within the monitored environment. Insider threats are, after all, about the actions of humans and not machines, and detection techniques will inevitably incorporate methods from the social sciences. But such data sets are difficult to come by. Researchers in this domain have two options—they can use or collect real user data, or they can use synthetic data. Because malicious insiders are, first and foremost, insiders, to collect real data, some organization must directly monitor and record the behavior and actions of its own employees. Confidentiality and privacy concerns create barriers to the collection and use of such data for research purposes. Thus, it is sometimes preferable to proceed with synthetic data.

With a mature synthetic data generation framework like the one used here, a user can flexibly control and rapidly and economically generate data sets with desired characteristics, size and quality relative to measurable characteristics. Because they are not real, the data sets are fully intact, with no need for de-identification, and free of privacy restrictions or limitations. Because they are generated, there is, theoretically, no limit on the length of time or number of individuals represented. Such data can be published, and thus allows other researchers to repeat experiments and compare algorithms. And because true positives in the data set are labeled, synthetic data can enable development, quality assurance, and performance testing in ways that may be difficult or impossible with real data.

Because using real, even de-identified, corporate data raises a variety of legal, ethical, and business issues, the DARPA Anomaly Detection at Multiple Scales (ADAMS) program turned to proxy data sets and synthetic data. Our task was to generate data to simulate the aggregated collection of logs from host-based sensors distributed across all the computer workstations within a large business or government organization over a 500 day period. Despite a widespread use of synthetic data to test classification systems [13], producing synthetic data that achieves a high level of human realism is a much more difficult problem [5]. In the process of performing this work, we made pragmatic choices to achieve sufficient fidelity and learned important lessons about the benefits and limitations of synthetic data in this domain. Our purpose in this paper is to give a simplified overview of our general approach, to highlight some of the challenges and lessons learned about the uses and misuses of synthetic data, especially regarding the role and meaning of realism in synthetic data, and to mention opportunities for future research.

II. Related Work

Most systems for generating cybersecurity test data focus on network traffic. These include network traffic generation appliances, such as BreakingPoint, and software solutions such as Swing [17] or Harpoon [15]. BreakingPoint is delivered with preloaded profiles for various types of traffic, while Swing and Harpoon are designed to learn traffic profiles by example. All three systems directly output network traffic to simulate a large number of machines. An alternate approach, used by Lincoln Labs' LARIAT [18] and Skaion's TGS, programatically drives applications on a virtual machine to more naturally generate the traffic. This approach is more interesting for our problem domain, since something like these systems could drive machines to generate other, host-based, sensor data. In fact, we see this in later work for DARPA ADAMS where Skaion's ConsoleUser, a relative of

DOI 10.1109/SPW.2013.37

TGS, is used to generate activity logs from a host-based sensor. Unlike these systems, the generator used for the effort discussed here simulates the sensor internally and directly outputs the activity logs.

Regardless of whether the final data is output directly by the generator, or is generated as a side effect of activity, LARIAT, TGS, and ConsoleUser follow an approach that calls for individual models for each agent being simulated. This model may be a set sequence of steps or a probabilistic model from which decisions are made. In the network domain, models and configurations based on real networks are already available. But the insider threat domain concerns itself with higher level user activity, and in this case, host-based activity, and such models are not readily available. Our work required us to choose reasonable dimensions and parameters for those dimensions in order to generate plausible background data.

While methods for generating large graphs that correctly model real-world network properties do exist ([8] [1]), they have not yet been integrated with a low-level agent-based system such as the one employed here in order to generate both the desired macro-level graph properties and the desired micro-level agent actions.

As an alternative to synthetic data, some data sets collected from real user studies are available. These were, however, unsuitable to support the breadth of detection techniques being developed for ADAMS. The Schonlau data set contains command line histories of 50 users and while useful for studying algorithms using that particular feature, it is missing important data types for insider threat detection including social structure, file accesses, content, and sentiment [14]. Ben Salem and Stolfo's RUU data set contains process, registry, and file access data from 18 users, along with those same data types from masqueraders [2]. Although we did make some use of this data, and it contains more data types than the Schonlau data, it is still missing several very important dimensions. Because of the difficulty of running such a study, both of these data sets are also based on a relatively small population of users. Previous work also addresses the structure and use of, as well as measures of realism for, synthetic data. Wright discuses the merits of synthetic data in running cybersecurity experiments which are controllable and repeatable [18]. Berk examines metrics for comparing real network traffic to traffic generated by LARIAT, and concludes that simulated traffic has substantial shortcomings [3]. Choosing measures appropriate to our domain we found that our synthetic data, though useful, also exhibited these substantial shortcomings.

III. Our Approach to Realism

Synthetic data will only ever be realistic in some limited number of dimensions, those being the ones that were called out as requirements, designed in and tested for. It will rarely be coincidentally realistic in any other dimensions. These dimensions of realism are normally defined by the System Under Test (SUT). By SUT, we refer to any procedure, process, or system which will use the data as an input. Thus, anything from a spreadsheet to an insider threat detection system that uses the data will contribute both criteria for realism and a judgment as to level of realism the data it finds. In the absence of such a clear context, ”realism” as an acceptance criteria cannot be measured, and the effort to achieve greater realism becomes open ended and unfocused. While consumers of synthetic test data often wish at the outset for realism in a large number of dimensions, the SUT's needs usually turn out to be limited to a much narrower set.

To illustrate the wide range of possible definitions for “realistic”, consider a hypothetical example of synthetic respondent data for the Census. Say that the Census mails out forms to people; people fill them in and send them back. Then the forms go through data capture, and the captured data gets processed in a variety of ways. If the data were intended to test handwriting recognition and data capture, then to be considered realistic, the data might only need to appear on forms in the correct locations. If the only criterion were that the captured data matched the data to be captured, the language content could be nonsensical and the data would still be sufficiently realistic.

If the SUT were to integrate data type validation rules, then data would need to not only be in the right position but also be semantically meaningful at least to the extent of having the correct data type. And so on. The more business logic the data capture system integrates, the more sophisticated the data needs to be to be considered sufficiently realistic to be useable. Once past data capture, the data might be processed in a variety of ways, each with their own additional filters and business logic. A single piece of data that may be valid on its own may be inconsistent in relation to other pieces of data. If the synthetic data were to reach deep enough through the system to reach every aspect to be tested, all those layers of consistency must be innate.

In our Census example, we see that if you want to test the system as a whole, you must create extremely rich and realistic data. But that if you focus is on some particular part of the process (e.g., only handwriting recognition), much of that richness and realism is of no value to you.

Providing data for insider threat research presents an unusual challenge for many reasons. One principle reason is that the threat domain is highly dependent on characteristics of human behavior as opposed to the more purely technical domains of network data generation or handwriting recognition. In order to provide a rich enough test bed for development and testing of techniques to identify those threats, the data needs to try to model the micro and macro effects of human agency. Furthermore, if consumers are to use the data not just for engineering, but for research, there is no SUT to dictate what constitutes sufficient realism. In this use case, it would be desirable, though probably not realistic, to use the data for *exploration* rather than simply *confirmation*. Put more concretely, synthetic data can be useful to confirm that a system detects a particular type of anomaly when that anomaly can be defined and measured. But without a much more sophisticated model of the natural world, it cannot be used to confirm that the given anomaly or collection of anomalies correlates to malicious insider behavior. At best, it can be used to confirm that the SUT can detect threats consistent with the data producers' models of malicious and normal user behavior. By way of analogy to the Census example above, data can clearly be generated to test whether the SUT can accurately recognize handwritten letters and numbers. But it would be difficult or impossible to generate data which could be used to sufficiently prove or disprove a SUT's ability to discern the census-taker's state of mind based on their writing.

Confirmatory Hypothesis: The SUT detects users who rank in the top 0.1% by frequency of removable USB drive use.

Exploratory Hypothesis: Users who rank in the top 0.1% by frequency of removable USB drive use are more likely to be malicious insiders.

Example 1: Confirmation vs. Exploration. Synthetic data is better suited to test confirmatory hypotheses than exploratory hypotheses.

While we may wish to be able to point to real data and say ”just like that”, one must articulate mesaureable and concrete requiremenets for realism. In this case, there were several different data consumers, each with different metrics and requirements for realism. There was, furthermore, a tension between the exploratory use of the data, where knowledge of the SUT could further invalidate any exploratory results, and the confirmatory use of the data, where knowledge of the SUT is absolutely necessary to provide useful test data. We resolved this tension by soliciting feedback from the detection teams about dimensions of realism that they would like to see added, or about unrealistic artifacts found in the synthetic data sets. Because the generator is parameterized and extensible, we were able to iteratively assess, extend, and tune its behavior.

When simulating human agents, as we do here, one must make a choice between generating observable variables directly or generating those observations implicitly by modeling latent variables. Restricting ourselves strictly to variables observable in our context is the simplest approach, and the one most easily provable and measurable against real data. For example, we might try to match the the average number of times per day an average real user copies file to a removable drive. Driving a system such as ours with latent variables means modeling the motivations, preferences, affinities, and other hidden states of the simulated users. Generating data from a latent model requires the generator designer to assert a theory of human behavior that goes beyond a simple count of observations, and the provability of these theories is limited by the availability of human subject research studies. We treaded a middle ground by combining models based on observable behaviors, models based on studies of human behavior, and models based on our own unvalidated but plausible theories of human behavior. By creating causal effects between these individually simple models, we are able to create a virtual world that is both consistent and seemingly complex. See Figure 1 below for an example of these interdependencies. In so doing, we aimed to provide synthetic data with sufficient realism and complexity to reasonably approximate real data for development and testing purposes.

IV. Implementation

Our implementation reflects these practical constraints and tradeoffs while yielding synthetic social network topologies and artifacts that are rich with a variety of types of link connecting sets of individuals and entities. It employs a number of different model types including graph models of the population's social structure, topic models for generating content, psychometric models for dictating behavior and preferences, models workplace behaviors, and others that don't fit neatly into these categories. The following are descriptions of some of the most important components of our implementation.

Relationship Graph Model. A graph representation of a real world relational network within organizations. This graph is constructed from a variety of typed/named interconnections representing relationships induced by organizational structure, subsets within the organization (departments), support relationships, and affinities derived from demographic elements. In early versions of the generator, the graph was limited to those affinities (a friendship graph) and in its simplicity, we were able to match several known power laws for social graphs [1]. But by adding all of these other elements to the construction of the relationship model, we lost the ability to implicitly induce those power law properties.

Asset Graph Model. A model of non-human assets such as computers, removable media, files, and printers associated with various individuals in the organization.

Behavior Model. A model dictating the how employees use assets, given their relationships, interests, affinities, and psychological profiles.

Communications Model. A probabilistic expression of the current state of organization relationships through a variety of electronic communications artifacts.

Topic Model. A user's topical interests both for data consumption (i.e., web browsing) and data production (i.e., email communication). In earlier versions of the generator, content was generated as a bag of words. The algorithm for choosing the mix of words is inspired by Latent Dirichlet Allocation (LDA) [4], which we postulated would be one of the analysis methods the data would be targeting. Since we were treating each synthetic web page or email as a collection of words about some topic or topics, we needed a labeled source for words associated with named topics. We chose a collection of Wikipedia articles as that source. Treating each article as a topic, we performed a TF.IDF [7] analysis to determine an approximate weighting of term importance for each topic. For each generated document, we first chose the mix of topics and the desired length of the document. Then we randomly selected the collection of words for each topic giving more weight to the terms with the highest TF.IDF scores. As the program progressed, it became more important to provide grammatically coherent text, so we partially abandoned this method of choosing individual words for a method of choosing complete sentences, in proportion to the desired mix of topics. It should be noted that these methods of generating text could only be useful for testing topic analysis algorithms. Other approaches, such as sentiment analysis, would require a different strategy.

Psychometric Model. Each user is endowed at the outset with a set of static personality characteristics as well as a dynamic job satisfaction variable. We chose these personality variables and their affects according to [10] and [6].

Decoy Model. A set of decoy files, based on Ben Salem and Stolfo's masquerade detection work [2].

Threat Scenarios. Each data set produced contains a small number of insider threat incidents, or scenarios. These scenarios were developed in consultation with counter-intelligence experts. Each scenario is instantiated as synthetic data in the same form and scope as the "normal" background data generated for all users in the dataset. For the high-level description of one of these scenarios, see Example 2.

"A member of a group decimated by layoffs suffers a drop in job satisfaction. Angry at the company, the employee uploads documents to Dropbox, planning to use them for personal gain."

This scenario would result in a number of observables in the generated data:

- Data streams end for laid-off co-workers, and they disappear from the LDAP directory.
- As evidenced by logon and logoff times, subject becomes less punctual because of a drop in job satisfaction.
- HTTP logs show document uplods by subject to Dropbox.

Example 2: Sample Threat Scenario

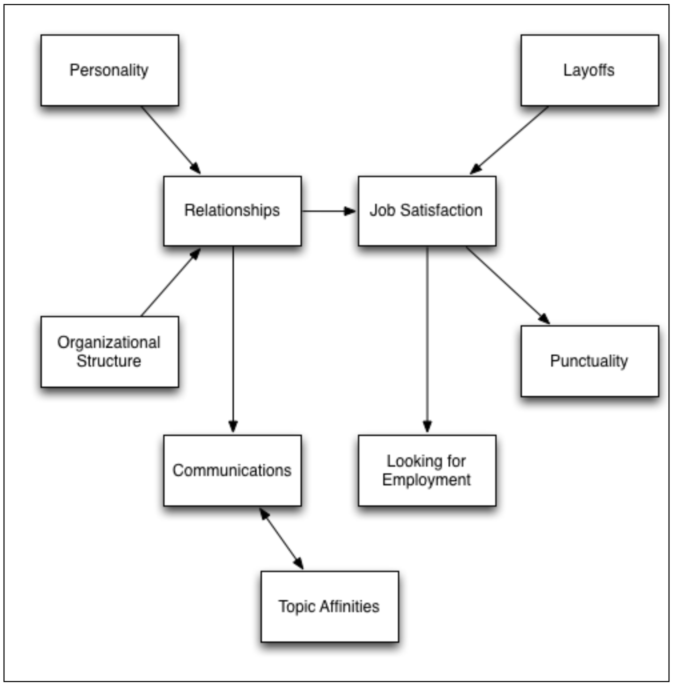

Figure 1: Example Variable Interdependencies. Read $A \rightarrow B$ as "A influences B".

Variable Interdependency. Causal relationships are established between various observable and latent variables, creating the appearance of a complex, natural system. For example, job satisfaction influences punctuality, graph relationships and communications affect topic affinities, and relationships to laid off employees affect job satisfaction. A simplified example appears in Figure 1.

Every decision point within each component is controllable, allowing both the tuning of each component (for instance, in order to establish the base rate for the presence of features or the actions of entities) and establishing triggers for, and injecting of, specific patterns or occurrences of features or the actions of entities.

The system collects information about both observable and latent state as data is produced. Both data and metadata are automatically captured for ground truth analysis.

V. Successes, Failures, and Lessons Learned

The data sets we produced were shared within a large community of researchers engaged in the DARPA ADAMS program to develop techniques for insider threat detection. Their analysis of our data confirmed our belief that simple models of social network topology, together with abstract models of user activity, can successfully aid in the development of those technologies, at least to the point of integration testing and confirmatory hypothesis testing. But their experience also showed that exploratory uses of this synthetic data are not fruitful, which could motivate the creation of more valid synthetic data in the future. The following are some of the most most important successes, failures, and lessons learned in this effort.

A. Development and Integration Testing

We confirmed that synthetic data is serves well as a stub during development and integration testing. It is an excellent fit for confirming that well-defined functions of the SUT work as expected.

B. No Privacy Restrictions

We confirmed the high value of data that is absolutely free of the confidentiality and privacy concerns that come with real data. Synthetic data provides the basis for meaningful experimental results and it can be used freely by researchers for demonstrations. Research can be performed anywhere, even by those who might otherwise be ineligible to access the real data. There is no exposure to possible privacy violations when using the data. Neither is there any need to de-identify the data, or worry that the data can be un-de-identified, because the data is not, nor never was, real.

C. Enhanced Collaboration

Another important benefit of synthetic data benefits to the insider threat research community is that it enables broad collaboration and increasing the velocity of innovation. Synthetic data provides a common basis for all researchers, allowing them to share a common set of threat scenarios. Because with synthetic data the ground truth can be perfectly known, researchers can readily explore data transforms and test algorithms. Enhanced collaboration enables design of common representations and interfaces that promote and enable integration of algorithms within and between groups and areas of research.

D. Controllable Social Network Topology

Research has shown that real-world social network graphs exhibit similar properties even when the graphs seemingly represent very different relationships (e.g., blog posts, campaign donations, and user ratings of movies). Academics interested in understanding these graphs have begun compiling ”laws” that real-world graphs tend to follow [1], [9].

While the research community found our generated data to demonstrate many important characteristics of realism, in our early data sets they also found that the time-varying social network extracted from our data sets lacked many of these expected laws found in real social networks.

While the laws of real-world networks do seem to be surprisingly consistent across real-world domains, the work used to derive the laws did not study the kinds of network we are simulating (i.e., a social network of friends within an office environment). Consequently, it may not be the case that our simulated networks should follow the laws of other networks. However, the recurrence of these laws across many different networks makes a compelling suggestion that our social network would be more realistic if it also followed these laws.

This shortcoming also illustrates another important benefit of synthetic data for insider threat research: controllable social network topology. With synthetic data, the social network topology can be both controlled and perfectly known. After further tuning, our later data sets were able to exhibit three of the social graph power laws that one would expect to find in a real social graph (Figure 2). As noted in Section IV, we achieved this result only with the friendship graph in isolation. In this network of friendships, we explicitly chose node degrees that would result in the expected properties. But the observed communication graph was composed of both this friendship graph and a corporate organizational graph, reflecting the corporate reporting structure and project assignments. Imposing those additional constraints created a composite graph that obscured the power law properties of the friendship graph.

Where graph topology is important, future work in this area should consider approaching the problem starting with an unlabeled graph exhibiting the desired properties, such as one produced by a Kronecker graph generator [8]. Semantics such as friendships and project assignments would then be assigned according to the topology, rather than attempting to produce a macro-level topology based on agent-level effects. While this approach presents its own difficulties, based on our experience here, it is an avenue worth exploring.

E. Text Mining

We had hoped to provide email, file, and web page content that would be valuable in some way to researchers developing topic and sentiment analysis algorithms. Achieving that goal turned out to be even more difficult than initially anticipated. To test simpler analysis algorithms, our LDA-based bag of words, and later bag of sentences, approach might have been sufficient. But the systems under test wished to perform more sophisticated analyses and would have benefited from more realistic content. Well-known methods such as those built upon context free grammars [16] and n-gram models [12] exist, but these may still have been insufficiently realistic and language generation systems are generally limited to specific problem domains. We believe improved generalized artificial text generation is an important area for future research if truly realistic insider threat data is to be generated.

F. Data “Cleanliness”

Though some inconsistencies were deliberately introduced, the synthetic data produced here turned out to be far more consistent and “clean” than the real data that the program eventually targeted. Our design choices about where to intentionally introduce inconsistencies, not surprisingly, failed to match the actual sources of inconsistency in the real data. We do not, however, regret our choice to

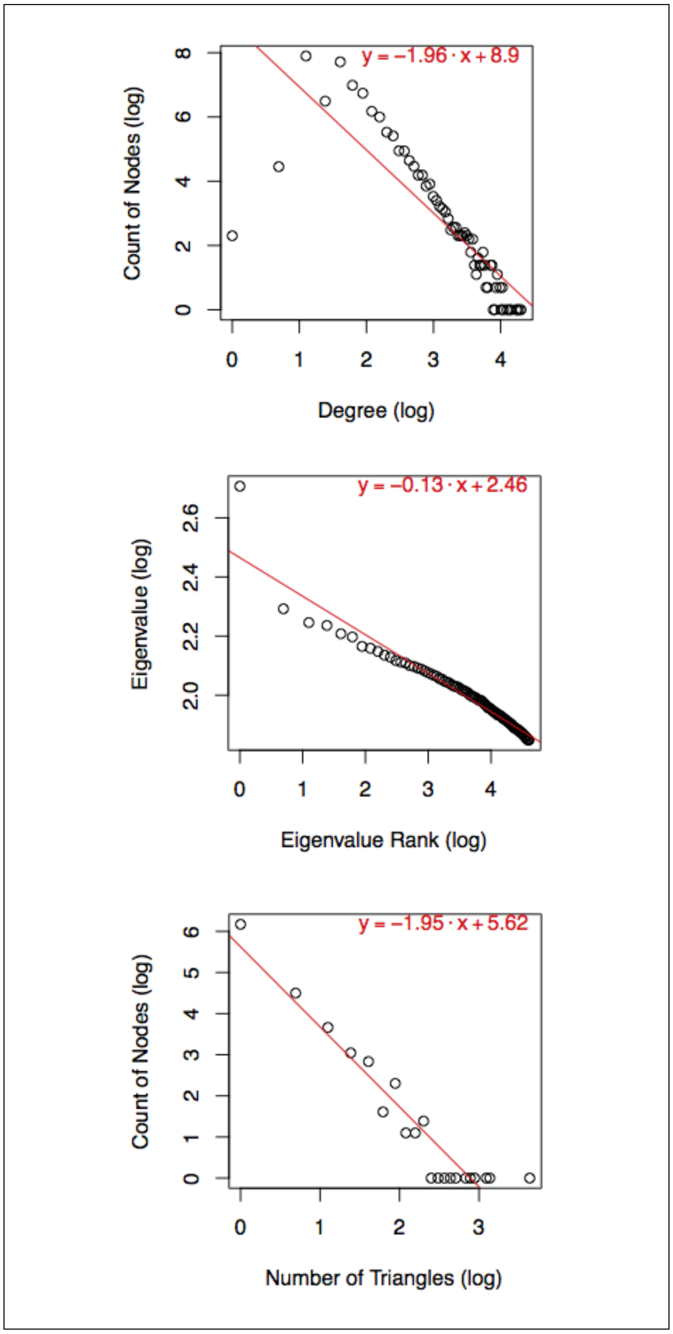

Figure 2: The friendship graph, in isolation only, conforms to the (top to bottom) node degree, eigenvalue rank, and triangle power law properties of social graphs.

insert intentional inconsistencies into the data, as it prepared the data consumers for the class of problems they would encounter with real data. Synthetic data can be unrealistic because it presents states that would impossible in real life, but it can also be unrealistic because it fails to take into account the inevitable flaws in the sensors collecting the data.

G. *Inability to Verify Theories*

Because we did not have access to analagous real, de-identified, data, our choices for parameter values were largely ad hoc rather than derived from such real data. (One notable exception is the behavior surrounding file decoys, which was based on statistics from a published user study [2].) As a result, the detection research teams were unable to use the data to confirm theories of how benign and malicious users behave. Instead, the data provided a plausible development and demonstration platform for the operation of the detection systems as a whole. To improve upon this clear deficiency in validity, future synthetic data sets for insider threats should strive to include both: 1. normal background data based on models and parameters derived from real data, and 2. threat activity based on technical logs of real insider threat activity. This goal is not easily attainable. Sources of (2) are rare or non-existent, but (1) combined with high confidence synthetic threat activity can still provide value in proving certain theories of insider threat detection.

VI. Conclusions and Future Work

Access to data remains a significant impediment to advancing the science of insider threat research. Manually created data sets don't provide the coverage and consistency needed to test complex systems and come with a very high price and error rate. Production data sets can't be disseminated without jeopardizing personally identifiable, proprietary, or even classified information. They are static and generally in limited supply, so it is very hard to get multiple data sets with matching known characteristics. De-identified data has elements masked that make it less effective for testing while not fully eliminating privacy/confidentiality risks. Random synthetic data generators create absurd data sets, useless for the type of sophisticated systems we contemplate here. Fully synthetic data can't replace real data, but, along with other benefits, it can significantly lower the barriers to entry into research requiring such data and provide the type of experimental control necessary to help establish a solid scientific foundation for such research.

However, the user of synthetic data must keep its limitations in mind. Though the data can be made quite complex, as we have shown here, it will only be realistic in those dimensions where realism is explicitly defined and measured. Unless those dimensions of realism are made to match rigorous models of the natural world, the ability of researchers to use synthetic data to prove theories of human behavior will be limited.

We see a number of areas for future research, some of which we are already pursuing:

- **Efficient Generation of Threat Scenarios**. Since the threat scenarios were authored individually, and the generator manually extended for each one, a relatively small number of truly different threats were produced. In ongoing work, we seek to automatically generate threats to provide a greater number and variety of test cases, as well as to implement an API to incorporate these automatically authored scenarios into the generator.

- **Better Models of Realism**. Continued study of models representing normal user behavior on computer systems as well as collection of baseline parameters from real users to drive and tune those models in synthetic data generators. By improving the quality of the innocuous background behavior, it becomes possible to more effectively hide threat signals. Having more of these formal models of realism also leads to *metrics* for realism, and we believe measurability is key to synthetic data progress and experimental validity.
- **Model More End-User Applications**. Enhance synthetic data by injecting applications and their artifacts. Allow malicious behavior to exploit the presence of new applications.
- **Improved Social Graph Generation**. Further exposure and tuning of the social graph generator to achieve both micro- and macro-level realism, and exploring the integration of other social graph generators (such as [8]) to extend capabilities.
- **Improved Content Creation**. Further improvement of natural language generation especially incorporating the expression of sentiment as part of content.

Acknowledgment

This material is based upon work funded and supported by the ADAMS (Anomaly Detection at Multiple Scales) Program of the Defense Advanced Research Projects Agency (DARPA) 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 Department of Defense. The findings and conclusions expressed are those of the authors and do not necessarily represent the views of the U.S. Government.

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

References

[1] Akoglu, Leman, and Christos Faloutsos. "RTG: a recursive realistic graph generator using random typing." Machine Learning and Knowledge Discovery in Databases (2009): 13-28.

[2] Ben Salem, Malek, and Salvatore Stolfo. "Modeling user search behavior for masquerade detection." Recent Advances in Intrusion Detection. Springer Berlin/Heidelberg, 2011.

[3] Berk, Vincent H., Ian Gregorio-de Souza, and John P. Murphy. "Generating realistic environments for cyber operations development, testing, and training." In Proceedings of SPIE, vol. 8359, p. 835908. 2012.

[4] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." The Journal of machine Learning research 3 (2003): 993-1022.

[5] Christen, Peter, *Data Matching*, Springer Verlag 2012, p. 179.

[6] Cullen, M.J., Russell, S., Bosshardt, M., Juraska, S., Stellmack, A., Duehr, E., and Jeansonne, K. "Five-factor model of personality and counterproductive cyber behaviors." Poster presented at the annual conference of the Society for Industrial and Organizational Psychology, Chicago, IL, 2011.

[7] Jones, Karen Sparck. "A statistical interpretation of term specificity and its application in retrieval." Journal of documentation 28, no. 1, 1972, pp. 11-21.

[8] Leskovec, Jure, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. "Kronecker graphs: An approach to modeling networks." The Journal of Machine Learning Research 11, 2010, pp. 985-1042.

[9] McGlohon, Mary, Leman Akoglu, and Christos Faloutsos. "Weighted graphs and disconnected components." Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008.

[10] Mount, M.K., Johnson,E., Illies, R. "Relationship of personality traits and counterproductive work behaviors: The mediating effects of job satisfaction." Personnel Psychology, 59, 2006, pp. 591-622.

[11] Newman, Mark E. J. "Power laws, Pareto distributions and Zipfs' law." Contemporary Physics, 46 (5):323351, 2005.

[12] Oh, Alice H., and Alexander I. Rudnicky. "Stochastic natural language generation for spoken dialog systems." Computer Speech & Language 16, no. 3, 2002, pp. 387-407.

[13] Paxton, K. Bradley, and Hager, Thomas. "Use of synthetic data in testing administrative records systems." Proceedings, Federal Committee on Statistical Methodology (FCSM), Washington, DC, 2012.

[14] Schonlau, Matthias. (2013, Feb. 1). "Masquerading user data" [Online]. Available: http://www.schonlau.net/intrusion.html

[15] Sommers, Joel, and Paul Barford. "Self-configuring network traffic generation." Proceedings of the 4th ACM SIGCOMM conference on Internet measurement. ACM, 2004.

[16] Stribling, Jeremy, Max Krohn, and Dan Aguayo. (2013, Feb. 1) "SCIgen-an automatic CS paper generator." Available: http://pdos.csail.mit.edu/scigen

[17] Vishwanath, Kashi Venkatesh, and Amin Vahdat. "Swing: realistic and responsive network traffic generation." IEEE/ACM Transactions on Networking, 17(3), 2009, pp. 712-725.

[18] Wright, Charles V., Christopher Connelly, Timothy Braje, Jesse C. Rabek, Lee M. Rossey, and Robert K. Cunningham. "Generating client workloads and high-fidelity network traffic for controllable, repeatable experiments in computer security." In Recent Advances in Intrusion Detection, pp. 218-237. Springer Berlin Heidelberg, 2010.