Generating Test Data for Insider Threat Detectors*

Brian Lindauer[1]†, Joshua Glasser[2], Mitch Rosen[2], and Kurt Wallnau[1]
[1] *Software Engineering Institute*
Carnegie Mellon University
Pittsburgh, Pennsylvania, U.S.A.
{lindauer, kcw}@sei.cmu.edu
[2] *ExactData, LLC*
Rochester, New York, U.S.A.
{joshua.glasser, mitch.rosen}@exactdata.net

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 hindrance. We outline the use of a synthetic data generator to enable research progress, while discussing the benefits and limitations of synthetic insider threat data, the meaning of realism in this context, comparisons to a hybrid real/synthetic data approach, and future research directions.

Keywords: insider threat, synthetic data, modeling and simulation

1 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, or data. Insider threats, the malicious acts carried out by these trusted insiders, include, but are not limited to, theft of intellectual property or national security information, fraud, and sabotage. Many government, academic, and industry groups seek to improve the dependability of their overall socio-technical systems by discovering and developing 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 families of options: those requiring use or collection of real user data, and those using only synthetic data. Malicious insiders are, first and foremost, insiders, and to collect real data, an 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 properties. Because they are not real, the data sets are fully intact, with no need for de-identification, and are free of privacy restrictions or limitations. Because they are generated, there is,

* A preliminary version of this paper appeared in the *Proceedings of the IEEE Security and Privacy Workshops* 2013 [1].

† Corresponding author: Software Engineering Institute 4500 Fifth Avenue, Pittsburgh, PA 15213, USA, Tel: +1-512-666-5438

theoretically, no limit on the length of time or number of individuals represented. Such data can be published, which allows other researchers to repeat experiments and compare algorithms. Also, 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 of obvious sensitivity reasons, we refer to the insider threat research that motivated our data synthesis project only as “the research program.” In light of the variety of legal, ethical, and business issues associated with using real—even de-identified—corporate data, the research 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 widespread use of synthetic data to test classification systems [2], producing synthetic data that achieves a high level of human realism is a much more difficult problem [3].

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.

2 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 [4] or Harpoon [5]. 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 [6] 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 the research program where Skaion’s ConsoleUser, a relative of TGS, is used to generate activity logs from a host-based sensor. This later work overlays synthetic data on real data, using a strategy similar to previous visualization data generation work at Pacific Northwest National Labs [7]. We discuss the relative merits of this related approach in the Comparison to Hybrid Test Data section of this report. 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 to generate plausible background data.

While methods for generating large graphs that correctly model real-world network properties do exist ([8] [9]), 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 for which the data was intended. 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 [10]. 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 [11]. 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 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 that are controllable and repeatable [6]. Berk examines metrics for comparing real network traffic to traffic generated by LARIAT and concludes that simulated traffic has "substantial shortcomings" [12]. Choosing measures appropriate to our domain, we found that our synthetic data, though useful, also exhibited these substantial shortcomings.

3 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 that 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 the level of realism of the resulting data. 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 United States Census. Say that the Census mails out forms to people, who 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 was 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 in order to be considered sufficiently realistic to be usable. Once past data capture, the data might be processed in a variety of ways, each with 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. Making synthetic data consistent in many dimensions and at many layers of abstraction is extremely difficult to do without making the consistency 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 if you focus on some particular part of the process (e.g., only handwriting recognition), much of that richness and realism will be 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. To provide a rich enough test bed for development and testing of techniques to identify those threats, the data needs 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* (see Example 1). 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 that 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 measurable and concrete requirements 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 average number of times per day an average real user copies a 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 3 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.

4 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 links 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 of workplace behaviors, and others that do not fit neatly into these categories. The following are descriptions of some of the most important components of our implementation.

Dynamic Data Generator$^{\text{TM}}$: ExactData's Dynamic Data Generator$^{\text{TM}}$(DDG) provides a generic capability for creating large data record collections consisting of multiple record types, with complex constraints and relationships within and across records, and with randomness (within the constraints) reflecting underlying population demographics. The current technology has been deliberately developed to provide a clear and well-defined mechanism to add to and extend our existing capabilities as limitations of the current capability are encountered. Data is rarely of interest simply in isolation; there is generally a greater context from which it derives significance and meaning, and without which interpretation is extremely limited. For each test data set being generated, DDG instantiates a model universe we call GAMUT (Great Automated Model Universe for Test), illustrated in Figure 1.

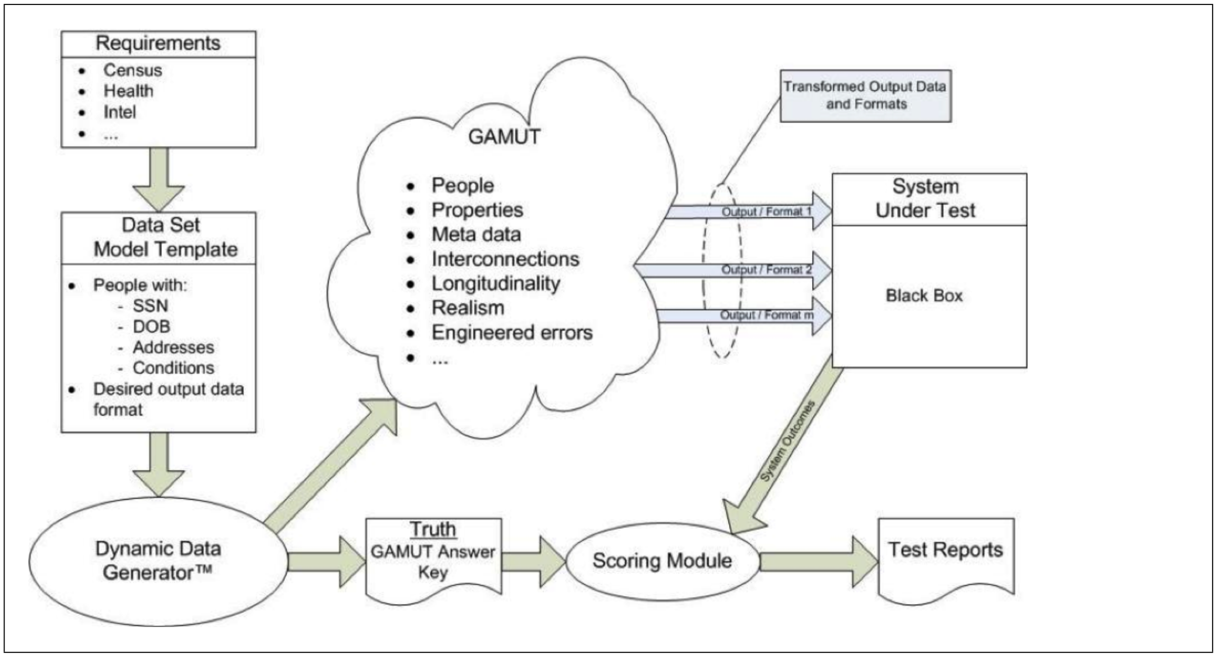


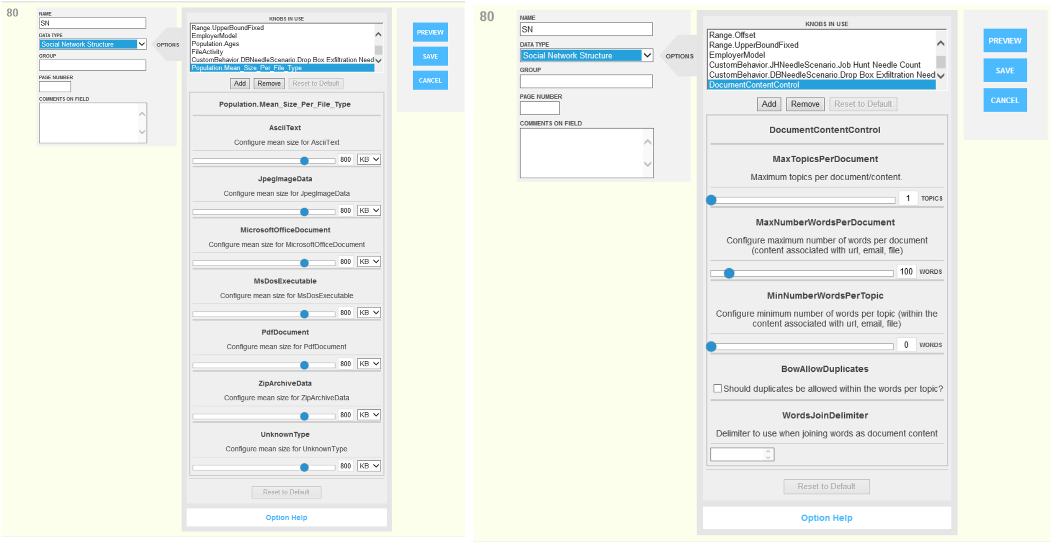
Figure 1: GAMUT - Great Automated Model Universe for Test

GAMUT contains first-principal models relevant to the data set being generated. It also supports numerous configuration controls so as to enable customization of the data generation logic according to the customer requirements. It gives the data the greater context the SUT requires for the data to be correct and meaningful, and to achieve the level of realism demanded. This approach also enables the creation of consistent disparate data stores—numerous output files that reflect the data from GAMUT in different formats. Also, it is precisely because of this design that DDG is capable of producing subtle engineered errors which are truly "needles in the haystack." Using our GAMUT approach, the synthetic threat behavior along with the artifacts generated by the behavior (the "needles"), within our insider threat data sets, are emergent from the initial properties assigned to the users. They are fully and naturally embedded within the behavior and trace artifacts from the general population of users (the "haystack"), without any necessity for a second pass over the generated data.

Once the generative models are established within DDG, the system exposes the supported configuration points and makes them accessible to the operator through a graphic user interface (see Figure 2). Examples of data set characteristics that can be configured include the following:

- including or excluding malicious insiders
- demographics
- personality and psychological factors
- social network topology
- organizational structures
- work processes
- file system artifacts
- the frequency and complexity of synthetic communications such as instant messaging, email, and documents

Data is generated by simulating these users' actions and interactions as they move forward in time.



NAME: SN
DATA TYPE: Social Network Structure
GROUP:
PAGE NUMBER:
COMMENTS ON FIELD:

Range.UpperBoundFixed
EmployerModel
Population.Ages
FileActivity
CustomBehavior.DBNeedleScenario.Drop Box Exfiltration Need
Population.Mean_Size_Per_File_Type

Population.Mean_Size_Per_File_Type

AsciiText
Configure mean size for AsciiText
800 KB

JpegImageData
Configure mean size for JpegImageData
800 KB

MicrosoftOfficeDocument
Configure mean size for MicrosoftOfficeDocument
800 KB

MsDosExecutable
Configure mean size for MsDosExecutable
800 KB

PdfDocument
Configure mean size for PdfDocument
800 KB

ZipArchiveData
Configure mean size for ZipArchiveData
800 KB

UnknownType
Configure mean size for UnknownType
800 KB

NAME: SN
DATA TYPE: Social Network Structure
GROUP:
PAGE NUMBER:
COMMENTS ON FIELD:

Range.Offset
Range.UpperBoundFixed
EmployerModel
CustomBehavior.JHNeedleScenario.Job Hunt Needle Count
CustomBehavior.DBNeedleScenario.Drop Box Exfiltration Need
DocumentContentControl

DocumentContentControl

MaxTopicsPerDocument
Maximum topics per document/content.
1 TOPICS

MaxNumberWordsPerDocument
Configure maximum number of words per document (content associated with url, email, file)
100 WORDS

MinNumberWordsPerTopic
Configure minimum number of words per topic (within the content associated with url, email, file)
0 WORDS

BowAllowDuplicates
□ Should duplicates be allowed within the words per topic?

WordsJoinDelimiter
Delimiter to use when joining words as document content

Figure 2: Sample DDG Configuration Screens

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 [9]. But by adding 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 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) [13], 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 [14] 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 [15] and [16].

Decoy Model: A set of decoy files that are based on Ben Salem and Stolfo's masquerade detection work [11].

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 data set. 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 uploads by subject to Dropbox.

Example 2: Sample Threat Scenario

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

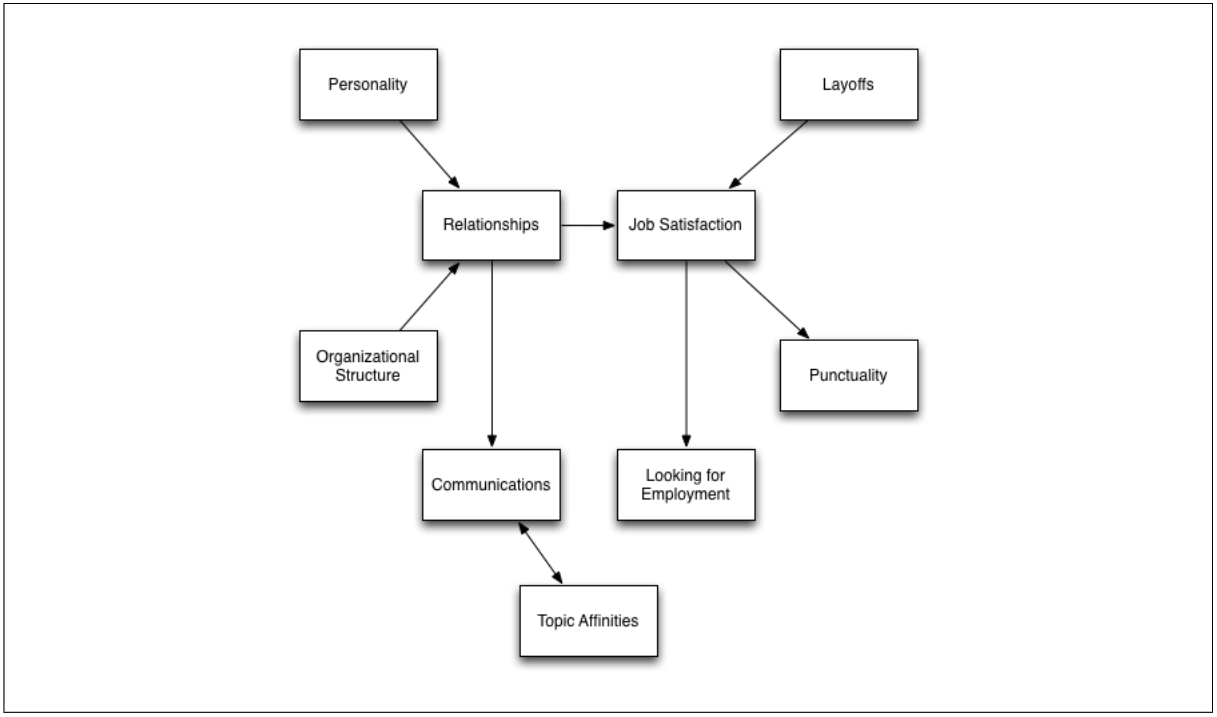


Figure 3: Example Variable Interdependencies. Read $A \rightarrow B$ as "A influences B."

to laid off employees affect job satisfaction. A simplified example appears in Figure 3.

Every decision point within each component is controllable, allowing both the tuning of each component (for instance, 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 meta-data are automatically captured for ground truth analysis.

5 Successes, Failures, and Lessons Learned

The data sets we produced were shared within a large community of researchers engaged in a research 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, and this 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.

5.1 Development and Integration Testing

We confirmed that synthetic data 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.

5.2 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 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 since the data is not, and never was, real.

5.3 Enhanced Collaboration

Another important benefit of synthetic data to the insider threat research community is that it enables broad collaboration and increases the velocity of innovation. Synthetic data provides a common basis for all researchers, allowing them to share a common set of threat scenarios. With synthetic data, the ground truth can be perfectly known; therefore, 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.

5.4 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 [17, 9].

While the research community found that our generated data demonstrated 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 (e.g., 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 (see Figure 4). As noted in Section 4, 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, it is an avenue worth exploring.

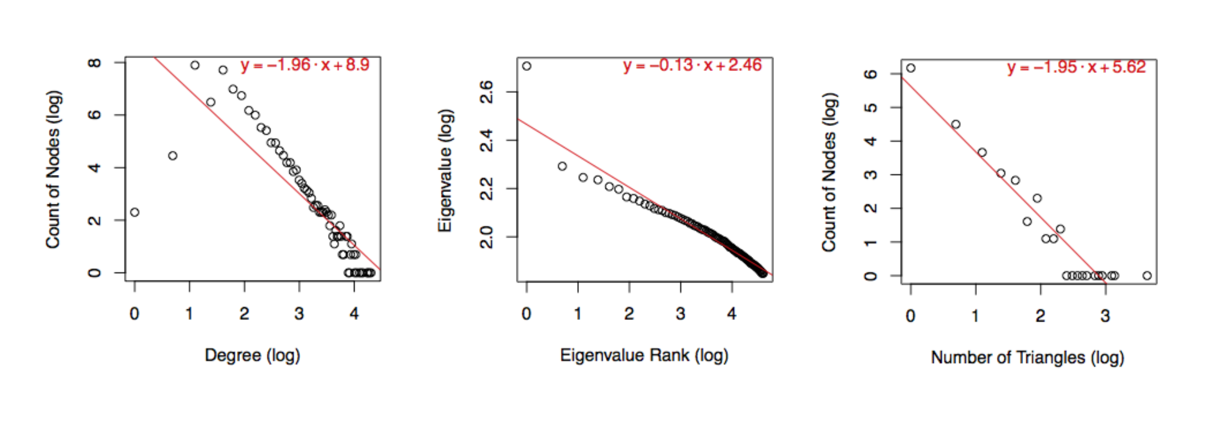


Figure 4: The friendship graph, in isolation only, conforms to the (top to bottom) node degree, eigenvalue rank, and triangle power law properties of social graphs.

5.5 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 [18] and n-gram models [19] 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.

5.6 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 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.

5.7 Inability to Verify Theories

Because we did not have access to analogous 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 [11].) 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.

6 Comparison to Hybrid Test Data

The data sets described above consist of synthetic background data integrated with synthetic incident data. This is only one of the four possible permutations of real/synthetic background and real/synthetic incident data that can form a test data set. For comparison, the Software Engineering Institute (SEI) provides "red team" data for test and evaluation of insider threat research. This red team data is a hybrid of *real* background data together with synthetic incident data. The real data is a de-identified copy of logs from host-based sensors and is augmented in place with synthetic incident data. In producing these data sets, we leverage a metaphor of drama and narrative. The real users are regarded as a pool of "user-actors" who are characterized and abstracted by a "central casting" service; user-actors are subsequently "cast" as characters in dramatic stories, which we refer to as "scenarios." The "real world" actions of user-actors recorded in a data corpus are then augmented by the synthetic actions of virtual characters recorded performing their dramatic actions. The resulting hybrid data is presented as a performance to an audience or analysts, as seen through the medium of the detectors under test. This approach combines the advantages of realism of monitored user behavior and experimental control of synthetic data to deliver test data that is high in realistic social complexity, low in technical and social artifacts, and targeted to the evaluation concerns at hand. The process is similar in many ways to a scenario-based data generation process used in previous work on visualization data synthesis [7].

Although hybrid data provides a more realistic environment for development and integration testing, the fact that access to it is highly restricted—in a closed lab in our case—makes using it in this fashion less practical than the fully synthetic alternative. As a result, unrestricted fully synthetic data is usually a more appropriate choice for this type of activity. However, as discussed above, fully synthetic data is not generally an appropriate basis for testing scientific theories. Hybrid data's real strength over purely synthetic data lies in the ability to verify such theories. Because all of the non-threat data comes from real users, researchers can draw scientifically valid conclusions about the base rates of human activity and social network topological properties in the monitored environment. A data set, in some sense, provides only a single point of reference for these properties, but this is no small advantage, especially in the case of insider threat detection, where approaches seek to differentiate a small number of malicious actors from a large and diverse population of benign users.

Confounds to scientific validity do exist even in hybrid data, though they are less obvious than in the fully synthetic case. As in fully synthetic data, the data producers exert full control over the actions of the simulated malicious insiders. The generalizability of a tool's ability to detect the inserted threats is largely dependent on the correspondence between the inserted insider threat scenarios and future real-world instances. We mitigate this potential confound through separation of roles. Counter-intelligence experts on a separate team in the CERT Division of the SEI provide scenarios, which are realistic, but fictional, accounts of insider threat cases. Frequently, elements of the stories are inspired by real cases, though for sensitivity reasons, real cases are never used directly. The experts writing these scenarios do so without detailed knowledge of the monitoring environment or properties of the users who will be "cast" into the roles in their story. Other team members take responsibility for editing scenarios—as minimally as possible—to fit the environment from which the real background data is collected. This gives us data that contains naturally occurring background activity, along with synthesized activity that leading insider threat experts have chosen to represent the types of insider threats that the tools would encounter in an operational deployment.

7 Obtaining the Data

The fully synthetic data produced for this project can be downloaded on the CERT website (http://www.cert.org/insider_threat/datasets).

8 Conclusions and Future Work

Access to data remains a significant impediment to advancing the science of insider threat research. Manually created data sets do not provide the coverage and consistency needed to test complex systems and come with a very high price and error rate. Production data sets cannot 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 cannot 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 automate the process of generating 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.

References

[1] J. Glasser and B. Lindauer, “Bridging the gap: A pragmatic approach to generating insider threat data,” in *Proc. of the 2013 IEEE Security and Privacy Workshops (SPW’13), San Francisco, California, USA*. IEEE, May 2013, pp. 98–104.

[2] K. B. Paxton and T. Hager, “Use of synthetic data in testing administrative records systems,” in *Proc. of the Federal Committee on Statistical Methodology (FCSM), Washington, D.C., USA*, January 2012.

[3] P. Christen, *Data Matching*. Springer, 2012.

[4] K. V. Vishwanath and A. Vahdat, “Swing: Realistic and responsive network traffic generation,” *IEEE/ACM Transactions on Networking*, vol. 17, no. 3, pp. 712–725, June 2009.

[5] J. Sommers and P. Barford, “Self-configuring network traffic generation,” in *Proc. of the 4th ACM SIGCOMM Conference on Internet Measurement (IMC 04), Taormina, Sicily, Italy*. ACM, October 2004, pp. 68–81.

[6] C. V. Wright, C. Connelly, T. Braje, J. C. Rabek, L. M. Rossey, and R. K. Cunningham, “Generating client workloads and high-fidelity network traffic for controllable, repeatable experiments in computer security,” in *Proc. of the 13th International Conference on Recent Advances in Intrusion Detection (RAID’10), Ottawa, Ontario, Canada, LNCS*, vol. 6307. Springer-Verlag, September 2010, pp. 218–237.

[7] M. A. Whiting, J. Haack, and C. Varley, “Creating realistic, scenario-based synthetic data for test and evaluation of information analytics software,” in *Proc. of the 2008 Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV’08), Florence, Italy*. ACM, April 2008, p. 8.

[8] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, “Kronecker graphs: An approach to modeling networks,” *The Journal of Machine Learning Research*, vol. 11, pp. 985–1042, 2010.

[9] L. Akoglu and C. Faloutsos, “RTG: a recursive realistic graph generator using random typing,” *Data Mining and Knowledge Discovery*, vol. 19, no. 2, pp. 194–209, October 2009.

[10] M. Schonlau, “Masquerading user data,” http://www.schonlau.net/intrusion.html, 1998, accessed: 2014-06-12.

[11] M. B. Salem and S. J. Stolfo, “Modeling user search behavior for masquerade detection,” in *Proc. of the 14th International Conference on Recent Advances in Intrusion Detection (RAID’11), Menlo Park, California, USA, LNCS*, vol. 6961. Springer-Verlag, September 2011, pp. 181–200.

[12] V. H. Berk, I. G. de Souza, and J. P. Murphy, “Generating realistic environments for cyber operations development, testing, and training,” in *SPIE Proc. of Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense XI, Baltimore, Maryland, USA*, vol. 8359, May 2012. [Online]. Available: http://dx.doi.org/10.1117/12.924762

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, March 2003.

[14] K. S. Jones, “A statistical interpretation of term specificity and its application in retrieval,” *Journal of Documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[15] M. Mount, R. Ilies, and E. Johnson, “Relationship of personality traits and counterproductive work behaviors: The mediating effects of job satisfaction,” *Personnel Psychology*, vol. 59, no. 3, pp. 591–622, Autumn 2006.

[16] M. Cullen, S. Russell, M. Bosshardt, S. Juraska, A. Stellmack, E. Duehr, and K. Jeansonne, “Five-factor model of personality and counterproductive cyber behaviors,” 2011, poster presented at the annual conference of the Society for Industrial and Organizational Psychology, Chicago, Illinois, USA.

[17] M. McGlohon, L. Akoglu, and C. Faloutsos, “Weighted graphs and disconnected components: patterns and a generator,” in *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’08), Las Vegas, Nevada, USA*. ACM, August 2008, pp. 524–532. [Online]. Available: http://dx.doi.org/10.1145/1401890.1401955

[18] J. Stribling, M. Krohn, and D. Aguayo, “Scigen–an automatic cs paper generator,” 2005, accessed: 2014-06-12.

[19] A. H. Oh and A. I. Rudnicky, "Stochastic natural language generation for spoken dialog systems," *Computer Speech & Language*, vol. 16, no. 3, pp. 387–407, July–October 2002.

Author Biography

Brian Lindauer is a Research Scientist in CERT's Science of Cybersecurity group. His research focus is on applications of machine learning and data mining to cybersecurity problems. Prior to CERT, Lindauer worked was a senior software engineer for the development of a the CounterStorm network anomaly detection system. Much of his work at CERT has centered around developing rigorous tests for anomaly detection systems.

Joshua Glasser is Director of Research at ExactData, where his focus is on the creation of specialized test data for BIG DATA systems. He has extensive experience developing tools and techniques for the creation of dynamic synthetic data. Prior to ExactData, he has worked for many different Defense Research and Imaging companies, including Honeywell, Kodak, and Xerox. He is the founder of the consulting firm Robo Imaging Research, and he has held various teaching positions, including at Clarkson University and the Rochester Institute of Technology.

Mitch Rosen is a Senior Imaging Engineer at ExactData, where his work focuses on design and implementation of algorithms supporting the derivation of domain specific synthetic data. Previously he was a Research Professor at the Rochester Institute of Technology, researching issues in high quality color imaging and non-traditional display and capture technologies. He earned his Ph.D. from RIT.

Kurt Wallnau joined the SEI in 1993 to advance the theory and practice of software engineering. His current interest is in the use of theories of drama and narrative to specify threat data for testing insider threat analytics. From 2010-12 Dr. Wallnau contributed to NSF's XSEDE program, first in defining its software and systems engineering processes and later as manager of software development. Dr. Wallnau led SEI research that combined software model checking, real-time analysis, and program generation to create software with predictable runtime behavior backed by verifiable evidence. He has also led SEI advanced technology research in automated production of proof-carrying code, and strategy-proof auctions for allocating tactical network bandwidth for data fusion.

This material is based upon work funded and supported by the Department of Defense 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.