SANDIA REPORT
SAND2019-12011
Printed October 2019

Sandia National Laboratories

Detailed Statistical Models of Host-Based Data for Detection of Malicious Activity

Erin C.S. Acquesta[†1], Guenevere (Qian) Chen[⋆2], Susan S. Adams[3], Ross Bryant[1], Jason J. Haas[1], Nicholas T. Johnson[1], Paul Romanowich[4], Krishna Roy[2], Mayuri Shakamuri[1], Michael R. Smith[1], Christina Ting[1]

[1] Sandia National Laboratories, Mission Analytics
[2] University of Texas at San Antonio, Electrical and Computer Engineering Department
[3] Sandia National Laboratories, Human Factors
[4] Gonzaga University, Psychology Department

† eacques@sandia.gov
⋆ guenevereqian.chen@utsa.edu

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

Available to DOE and DOE contractors from

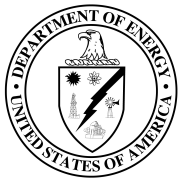
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods



ABSTRACT

The cybersecurity research community has focused primarily on the analysis and automation of intrusion detection systems by examining network traffic behaviors. Expanding on this expertise, advanced cyber defense analysis is turning to host-based data to use in research and development to produce the next generation network defense tools. The ability to perform deep packet inspection of network traffic is increasingly harder with most boundary network traffic moving to HTTPS. Additionally, network data alone does not provide a full picture of end-to-end activity. These are some of the reasons that necessitate looking at other data sources such as host data. We outline our investigation into the processing, formatting, and storing of the data along with the preliminary results from our exploratory data analysis. In writing this report, it is our goal to aid in guiding future research by providing foundational understanding for an area of cybersecurity that is rich with a variety of complex, categorical, and sparse data, with a strong human influence component. Including suggestions for guiding potential directions for future research.

ACKNOWLEDGMENT

We want to thank everyone that supported this exploratory effort that brought together a great team from multi-disciplinary backgrounds:

- The participants that agreed to have the Windows Logging Service installed on their computers, collecting a variety of host-based data information, as well as taking the time to fill out the surveys to help us initiate our exploration into the correlations between baseline human behaviors with patterns of behaviors with their computer systems,
- Dr. Travis Bauer, at Sandia National Laboratories, for initiating the collaboration, introducing the PIs, and offering guidance and support throughout the project,
- Dr. Shouhuai Xu, at The University of Texas at San Antonio, for his consultation and expertise,
- Ryan Holt and Daniel Garcia for their expertise and insight during the data exploration phase and our research, and
- Dr. Courtney Dornburg for her guidance and inspiration in diving down into the science of cognitive psychology.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Increasingly, advanced cyber defense analysis is turning to host-based data to use in research and development to produce the next generation network defense tools. The detailed information collected on the host regarding process and user interaction can provide context for determining malicious activity. In our multi-disciplinary research effort, we collaborated with experts from Sandia National Laboratories (SNL), the University of Texas at San Antonio (UTSA), and Gonzaga University with domain expertise in areas related to cybersecurity, psychology, and mathematics to bring together the knowledge needed to understand and infer behavioral patterns from host-based data. This report outlines our investigation into the processing, formatting, and storing of the data along with the preliminary results from our exploratory data analysis. This report is not meant to provide a comprehensive solution to any one particular problem related to host-based data. Instead, this report it meant to guide future research by providing foundational understanding for an area of cybersecurity that is rich with a variety of complex, categorical, and sparse data, with a strong human influence component.

1.1. Motivation

1.1.1. Collecting and Analyzing Host Logs

The ability to perform deep packet inspection of network traffic is increasingly harder with most boundary network traffic moving to HTTPS. Additionally, network data alone does not provide a full picture of end-to-end activity that occurs for an event. These are some of the reasons that necessitate looking at other data sources such as host data.

Comprehensive defensive and cybersecurity solutions require a robust understanding of the underlying statistics of the data. Researchers should first understand how statistically significant an indicator needs to be to reliably differentiate abnormal and unsafe behavior from normal background variance. Therefore, developing and testing a set of comprehensive statistical analyses of host-based data, including time series and variance analyses would set a foundation for future research, helping us to understand how host-based data varies over time naturally (such as the day/night cycles) and the kinds of features that may contain the most information regarding human interaction and the underlying semantics.

The very few existing and public cybersecurity relevant datasets have many limitations as summarized in Los Alamos National Laboratory's (LANL) recent publication [13]. For example, many datasets are collected from specific and pseudo real-world events rather than daily operational environments. Most datasets are synthetic and created using models intended to represent specific phenomenon of relevance. Some commonly used datasets are egregiously outdated. Underlying systems, networks and attacks represented in some outdated datasets are 30 years old, which can no longer represent cyber phenomenon under modern computing environments.

Specific to the LANL open-sourced host-based dataset, there are 20 events captured from LANL's 13,000+ hosts located in their operational network recorded over 90 days. These events from the

host logs included in LANL's dataset are only related to authentication and process activity on each machine. As discussed by the authors "it is important that researchers have a thorough understanding of the context, normalization processes, idiosyncracies and other aspects of the data" [13].

While existing dataset have been useful in allowing the cybersecurity research community to start to drive discussions in understanding host-based data, we found it necessary to collect our own data. This allows us to expand upon this understanding and start to answer the questions that came from the analyses of those earlier datasets. Therefore, we recruited 36 participants who are full-time staff working at a major U.S. university. These participants agreed to let us collect host logs from their university-owned work computer for 90 days for each participant. More than 400 EventIDs occurred (e.g., events related to file system, registry, session, device-related events). With this new data collection, we expect to develop cyber security solutions moving beyond signature-based methods for advanced cyber threat detection and prevention. We provide a more detailed description of host-based data in Section 3.

1.1.2. Multi-disciplinary and Human Subject Research

Humans are the weakest link of cybersecurity [3, 5, 6]. Unintentional or inadvertent human mistakes can trigger the same regulatory obligations as cyber attacks [8]. On the other hand, humans are much more reliable than current cybersecurity solutions to spot and report the most disruptive breaches or attacks [14]. The human factor therefore is a double-edged sword, which motivates us to conduct this multi-disciplinary research to (1) understand human errors in cyberspace, (2) find relationship between host-based data and psychological principles of personality.

The current cybersecurity research community still focuses on developing technology solutions to secure cyberspace. Our long term goals regarding the human factor of cybersecurity research are (1) determine the root cause of human error, related to cybersecurity incidences and (2) how to monitor, measure, detect and reduce human error in time.

2. MULTI-DISCIPLINARY RESEARCH FOR CYBERSECURITY

"In many intellective tasks groups consistently outperform individuals [12]".

In our research, we have adopted this philosophy and chosen to emphasize that the group we put together should complement each other with each respective domain expertise. In this section, we provide a summary from each of the cybersecurity, psychology, and mathematical perspectives.

2.1. Cybersecurity Domain Expertise

Host-based data contains bountiful information for the cybersecurity community to develop technological solutions to protect the cyberspace. For instance, consider the following.

(1) Host-based data can be used to facilitate malware forensic efforts. Events that deviate from normal behavior would alert information technology (IT) operators that these events might indicate an initial attack vector (e.g., external threat or insider threat). These events and the correlation between a series of events are unique patterns for IT operators to discover new threats that bypass traditional detection mechanisms. Identification is the first step to remove malware and prevent further compromises.

(2) Host-based data can also be used to determine the breadth of a compromise in an operational enterprise network. Traditional intrusion detection systems detect anomalies based on network traffic. Using host-based data, IT operators could compare network data that indicates potential compromise with host-based data to confirm or refute the previous evidence.

2.2. Behavioral and Cognitive Psychology

Psychology can profitably aid cybersecurity by formulating and applying well-validated behavioral and personality measures, findings, and theories to identify who may be more or less likely to fall prey to cybersecurity vulnerabilities (e.g., malware). As described in Section 4, these measures suggest that there are certain general behavioral patterns and personality traits that are stable within an individual. We have focused on one stable measure, impulsiveness, to determine whether it makes individuals more or less vulnerable to cybersecurity issues as shown through host-based data. For future research, prominent theories and findings from cognitive psychological research (decision making, attention, memory, etc.) will be leveraged to further explore the tie between cybersecurity and the human factor.

2.3. Mathematics and Statistics Principles

The fields of mathematics and statistics are so closely related that it is very common to consider them one and the same, when in fact, a principled approach from each discipline focuses on a distinct component of predictive science. A principled mathematical approach will prioritize validation and verification of our abstract model formulation for phenomenons in the real world. Alternatively, as a data-driven discipline, a principled statistical approach emphasizes the effective implementation of the design of experiments for which we build inference about the world around us. In the paradigm of predictive science, as illustrated in figure 2-1, there is a direct implementation between our models and experiments, in addition to numerical simulations, that we leverage to derive targeted quantities of interest [11]. At each stage of our analysis, we introduce distinct sources of uncertainty as we make more assumptions, simplification, and discretization to derive solutions.

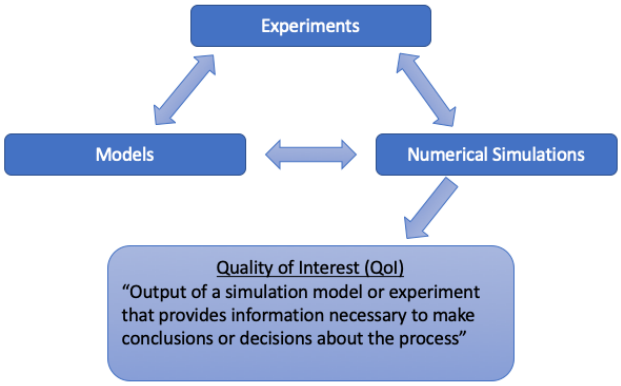


Figure 2-1 Predictive Science Diagram

Predictive science can aid cybersecurity analysis informed by the human factor to identify various sources of uncertainty to assess the confidence on resulting qualities of interest (e.g. output of a simulation model or experiment). In future research we will focus on the following: (1) how both epistemic (statistical, stochastic, irreducible) and aleatoric (systemic, reducible) uncertainty manifest in our analysis, (2) identification of which component of predictive science these sources of uncertainty are inherent, and (3) determine how we quantify, aggregate and propagate uncertainty to measure the overall uncertainty of our model predictions. This approach will result in a metric of model confidence for the assessment on the inference obtained about the cybersecurity systems.

Specific to the cybersecurity research we are outlining in this report, we can identify a number of sources of uncertainty. These include but are not limited to human variability and system outages. As we continue to integrate our multidisciplinary efforts, we will continue to quantify these sources of uncertainty and compare our findings with the expectations from subject matter experts (SMEs) in cybersecurity as well as psychology.

3. HOST-BASED DATA

The host-based data collected in this project contains all host event logs collected from 36 participants' computers running Microsoft Windows 10 on a large enterprise network for 90 days. The collected raw logs are pre-processed, anonymized, and saved in JSON format in order to preserve the structure of the original events. This study is approved by the Institutional Review Board (IRB), and informed consent is obtained before survey initiation and data collection by all participants.

3.1. Host-Based Data Analysis

Host-based data are records of computer events, either triggered by a user or by a running process. In this project, host logs are collected from all computers running Microsoft Windows with Windows Logging Service (WLS) [2]. WLS is a Windows service application installed on each host that forwards host event logs [13], along with the enterprises defined contextual data, to collection servers.

3.2. Host Logs Collection

3.2.1. Participant Recruitment

To recruit participants, we emailed project flyers to full-time administrative staff and their supervisors at a major U.S. university. This study targets full-time staff primarily because (1) the data collection began in the summer semester, and (2) this group of participant's host behavior are more predictable than students and faculty. Students and faculty host behavior is different in the summer compared to the rest of academic year. Full-time staff, however, is a group of users whose host behaviors are influenced very little by semesters. Additionally, administrative staff normally accesses limited, common and similar software or applications. However, faculty and students from different departments may use various software and some are rarely used by the public.

The 36 participants, whose ages range between 25 and 63 years (mean = 40.7, standard deviation = 12.0) with 72.2% female and 27.8% male. 42% of participants self-identified as Latino/Hispanic, 39% as Caucasian, and 19% as either Asian, African American, or American Indian.

3.2.2. Windows Logging Service

"The Windows Logging Service (WLS) is a Windows service that provides enhanced operating system information via standard syslog messages to any syslog format compatible log server. WLS augments traditional logging and forensic analysis with real-time reporting of contextual operating system (OS) information. WLS reads and sends all Windows event logs and adds extra data relevant to cyber security, such as cryptographic hashes and file metadata. In addition to

event logs, WLS monitors a variety of system details that are often cited in incident reports as IOCs. [4]"

We installed WLS on each participant's desktop work computers, and separately set up a data collection server. We created five virtual machines on a physical server to receive and distribute 36 host logs and avoid a single-point of failure. The raw data collected by WLS contains both network traffic and host logs. We saved raw data in the JSON format.

3.3. Data Pre-processing

This project aims at collecting and analyzing host-based data. Therefore, we first separated network traffic from host logs, anonymized host logs, and filtered logs without the *EventID* field. We then saved and encrypted the network traffic for future research.

We indexed the pre-processed host-based data in Elasticsearch [1] for data storage, and Kibana is used for data access, analysis and visualization. We provide a detailed introduction of the data analytics tools in Section 5.

3.4. Host Log Event IDs

WLS event logs contain information about each executed process with user-defined parameters, contextual data, and have a specific EventID. For example, the EventIDs of a successful user account logon/logoff are 4624/4634.Process starts and exits are recorded with EventIDs 4688 and 4689, respectively.

In our initial analysis of 10 weeks of host logs, we found more than 440 EventIDs. These events are initiated by users or computer processes, and some events are a consequence of successive response of related processes. For instance, a sequence of EventIDs 4624 $\rightarrow$ 4627 $\rightarrow$ 4688 $\rightarrow$ 4670 $\rightarrow$ 4689, mean (1) successful logon (4624); (2) user account control using group policy (4627); (3) a process (chrome.exe—Google chrome browser) starts (4688); (4) permissions on an object are changed by the process "chrome.exe" (4670); (5) the process "chrome.exe" has exited (4689).

We provide a list of the 20 most frequently seen EventIDs in our dataset and the meaning of each EventID in Table 3-1

Table 3-1 Rankings of the top 20 events out of 440 and their EventIDs and Description

EventID	Description
4703	A user right was adjusted
5447	A Windows Filtering Platform filter has been changed
4688	Process start
4689	Process end
4670	Permissions on an object were changed
4957	Windows Firewall did not apply the following rule
4624	An account was successfully logged on
4627	Group membership information
4672	Special privileges assigned to new logon
4945	A rule was listed when the Windows Firewall started
4799	A security-enabled local group membership was enumerated
7036	The <service name> service entered the <star/stop> state
4634	An account was logged off
1	Process creation
916	The beta feature EseDiskFlushConsistency is enabled in ESENT
4904	An attempt was made to register a security event source
4905	An attempt was made to unregister a security event source
4798	A user's local group membership was enumerated
4611	A trusted logon process has been registered with the Local Security Authority
6416	A new external device was recognized by the system

4. BEHAVORIAL PSYCHOLOGY ANALYSIS

As we initiated the analysis from a behavioral psychology perspective, we ran a preliminary study on impulsivity and delayed discounting to assess the potential exposure to "risky" behaviors that might be expected of the 36 participants.

4.1. Introduction

Impulsiveness is a multidimensional psychological construct that describes when an individual cannot inhibit a response, doesn't plan before an action is taken, takes unnecessary risks, or has difficulty delaying gratification. This last dimension of impulsiveness (gratification delay) has been extensively studied through the lens of problem behaviors, especially addiction. Individuals who prefer smaller, sooner rewards to larger, delayed rewards discount value more steeply than individuals who prefer larger, delayed rewards. This metric of "delay discounting" is significantly associated with several addictive behaviors, such as alcohol, cigarette and cocaine use, where individuals using these substances discount hypothetical monetary rewards more steeply than non-using individuals. Researchers have begun describing steep discounting as a trans-disease process, given the ubiquity of this association between substance use, other health-related problems (e.g., overeating, gambling) and steep discounting rates. This idea of delay discounting as a trans-disease process motivated the current research program, which expands the scope of problem behaviors to cybersecurity vulnerabilities. Our main hypothesis regarding these problem behaviors and delay discounting is that participants who show computer use patterns that make them more vulnerable to cybersecurity attacks will also be more likely to choose smaller, sooner rewards in a standardized delay discounting task.

4.2. Impulsiveness Measures

Delay discounting has traditionally been measured by a titrating procedure whereby an individual's choice causes the hypothetical reward amount to change until an indifference criterion is met for each delay (e.g., 1 day, 7 days, 30 days, etc.). However, for larger groups of participants, a 27-item delay discounting task with fixed hypothetical amounts and delays for each choice has shown equivalent and reliable results. The current studies used this later 27-item task designed by Kirby *et al.* [9]. The primary dependent variable is the k-value, which corresponds to the number of smaller, sooner reward choices made. The k-value is based on fitting a hyperbolic equation to the choices made.

In addition, we also measured another impulsiveness construct, risk taking, via the balloon analogue risk task (BART) [10]. The BART is a computerized task which displays a small simulated balloon accompanied by a balloon pump. Each click on the pump inflates the balloon 1° (about 0.125 in. [0.3 cm] in all directions). With each pump, $0.05 is accrued in a temporary reserve. When a balloon is pumped past its individual explosion point, a "pop" sound effect is generated from the computer and all money in the temporary bank is lost. The next uninflated balloon then appears on the screen. At any point during each balloon trial, the participant can stop

pumping the balloon and click the Collect $$$ button. Clicking this button transfers all money from the temporary bank to the permanent bank. Each of three different balloon colors has a different probability of exploding. The primary dependent variable is the average adjusted pumps (only for unexploded balloons) during the last 20 trials for the balloon color with the lowest risk (most pumps to potential explosion).

4.3. Preliminary Results:

Given that delay discounting and risky choices are separate impulsiveness constructs, we first calculated a Spearman's rank-order correlation on *k*-values (delay discounting) and mean adjusted pumps (BART) for the 30 participants that completed both tasks. Results showed a negative, but non-significant relationship, relative-significance $(rs) = -0.19$, probability $(p) = 0.33$, suggesting these two constructs are not related in the current sample.

We tested delay discounting internal consistency by using an automated scorer developed by Kaplan, *et al.* to asses choice consistency (internal reliability) [7]. Overall consistency ranged from 0.85 - 1.00. Scores below 0.75 typically indicate questionable consistency.

Internal consistency for the BART was assessed by Spearman's rank-order correlations between the mean adjusted pumps for each of the three balloon colors. Significant positive correlations were found between red and green balloons, $rs = 0.36$, $p < 0.05$ and green and blue balloons, $rs = 0.63$, $p < 0.01$, but not between red and blue balloons, $rs = 0.31$, $p = 0.09$.

We assessed construct validity for the impulsiveness tasks by correlating each task's dependent measure with self-reported neuroticism scores. Neuroticism is one of five stable and unique personality traits measured by the brief Ten-Item Personality Inventory (TIPI). One aspect of Neuroticism is Impulsiveness. Delay discounting *k*-values were negatively correlated with Neuroticism, $rs = -0.23$, but at a non-significant level, $p = 0.19$. Mean Adjusted pumps for the BART were positively correlated with Neuroticism, $rs = 0.35$, but only at a trend level, $p = 0.06$.

Once cybersecurity user data is available, we will conduct a media-split of all participants' data based on both delay discounting *k*-values and mean adjusted BART pumps. Means for behaviors that make individuals more vulnerable to cybersecurity attacks will be compared between low- and high-discounting groups (and low- and high-BART pumps) to determine whether participants who show computer use patterns that make them more vulnerable to cybersecurity attacks will also be more likely to choose smaller, sooner rewards in a standardized delay discounting task (or risky choices via the BART).

5. EXPLORATORY DATA ANALYSIS

5.1. Introduction

We were given approximately 12TB of Windows Logging Service log data that contained about 300M logged events. Each logged event has a unique EventID corresponding to the type of event that is logged, as described above in Section 3.4. Each event, contains a timestamp, ProcessID and UserID. We stored each event and all of its associated metadata as key/value pairs in JSON format, largely as ASCII text. This data volume and structure required a solution to be able to rapidly explore, generate statistics and reason about the data. Ultimately, we made the decision to index the data in an Elasticsearch instance at SNL [1]. This provided a storage solution with easy keyword search because of its underlying reverse index. On top of Elasticsearch, we used Python in Jupyter notebooks to programmatically access the Elasticsearch API and Kibana [1]. The Kibana interface is a GUI front-end to Elasticsearch, and it allowed us to rapidly explore the data. We broke into four teams in order to be able to explore different areas in the data with different approaches. Below we present the work of each team.

5.2. Team One

We started in Kibana with the Discover tab, which allowed us to view the raw JSON for a single event. This is extremely useful to understand individual fields and the types of value that they contain. By inspecting a single datum, we can make guesses as to which groups of fields might yield insight. For example, EventID 4703 has a ComputerName field and a DisabledSecurityPrivilegeList field and pairing these together in a quick summary informed us what security privileges have been disabled on a particular machine.

Our next step in Kibana was then to pivot to the Visualize tab and create a heat map visualization, which you can see in Figure 5-1. Once we got the visualization parameters tuned to our liking, we put multiple visualizations on a dashboard that could quickly summarize key features of the data that we have discovered. The Dashboard tab in Kibana allowed us to display multiple visualizations that update in real-time as we continued to analyze, drill-down, and pivot into the data. Iterating via this three-step Discover/Visualize/Dashboard method, we could rapidly experiment with presenting this high-dimensional data in a human-digestible format that helps us learn what the data contains. This was a necessary first step before we can ask and answer meaningful questions about the data.

The first thing that we noticed was that the data has a serious class imbalance problem. Of the 300M+ total events, 250M Events were from a single EventID (4703). The next most frequent was 25M events from EventID 5447. Each remaining logged Event had >=2M per unique EventID. So, we created dashboards for each of the two largest EventIDs to study on their own.

Data parsing errors were also evident in at least one case where the DisabledPrivilegeList (event 4703 data) field value contained binary data when it should have been completely plain text. Some of the 9 common fields across all data divided neatly into a few categories. For example,

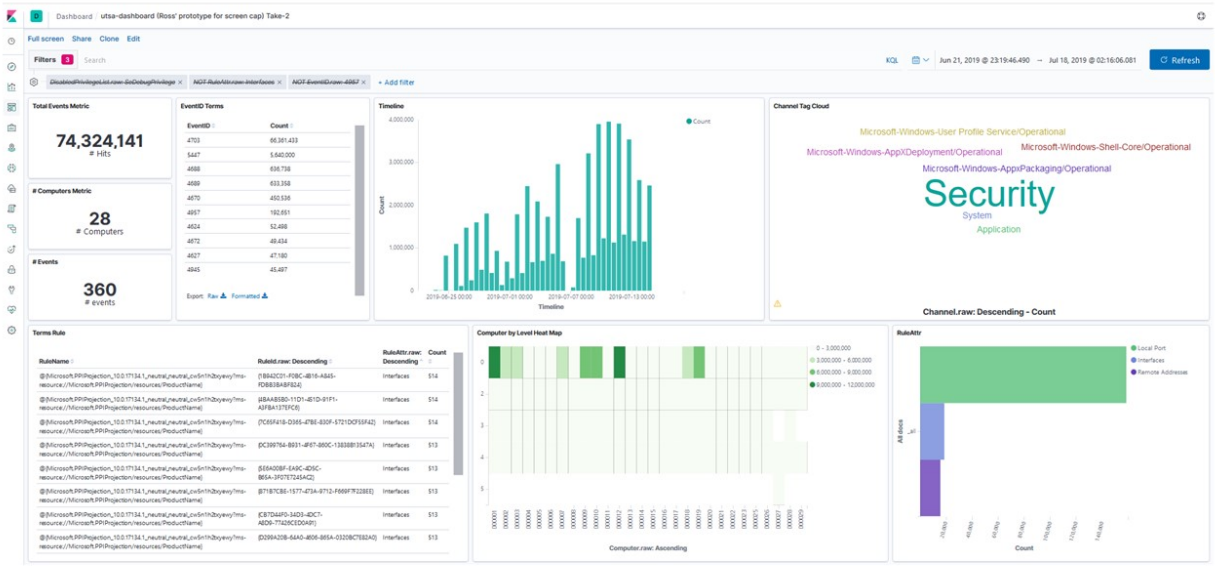


Figure 5-1 Kibana Dashboard Showing High Level Overview of the Data

Windows gives each logged event a Level from 0-5, 0 being common or benign and 5 being rare, severe or otherwise notable. Another common field was Channel which was only three categories: Application, System, and Security. We discovered that the Security Channel event logs have interesting information, such as the specific command line code that was executed.

5.3. Team Two

Another approach was to analyze which parts of the data could provide indicators of "importance". We could then continue digging into the Provider Names and Levels fields to gain insights into the timing of the events. The Provider Name field recorded the value that gave a high-level description of the vendor or software product that was doing the logging. The Level field indicated the severity of the log. The logic here was to see what was normal, which providers or logs could possibly be filtered out, and which ones might correlate with more interesting events.

5.3.1. Provider Name Insight

As of 9/5/2019, there are 181 unique values for Provider Name.

Microsoft-Windows-Security-Auditing is the most common Provider Name with 304,456,261 records.

- A security audit is a systematic monitoring of the security of a company's information system by measuring how well it conforms to a set of established criteria. Windows security auditing is a Windows feature that helps to maintain the security on the computer and in corporate networks.
- All of the Microsoft-Windows-Security-Auditing records have a level of 0, which is the least severe. To note, these 304,456,261 records represent approximately 99.9965% of all of the level 0 records.

Several of the other Provider Names seemed to deal with backing up systems and night guards (e.g., killing processes that ran after a certain time).

5.3.2. Level Field Insight

All of the documents in the current set have levels: 0-5. We provide their frequency count in Table 5-1.

Table 5-1 Level Field Frequencies

Frequency	Level
304,467,353	0
326,185	4
32,692	2
7,188	3
23	1
1	5

5.3.3. Additional General Insight

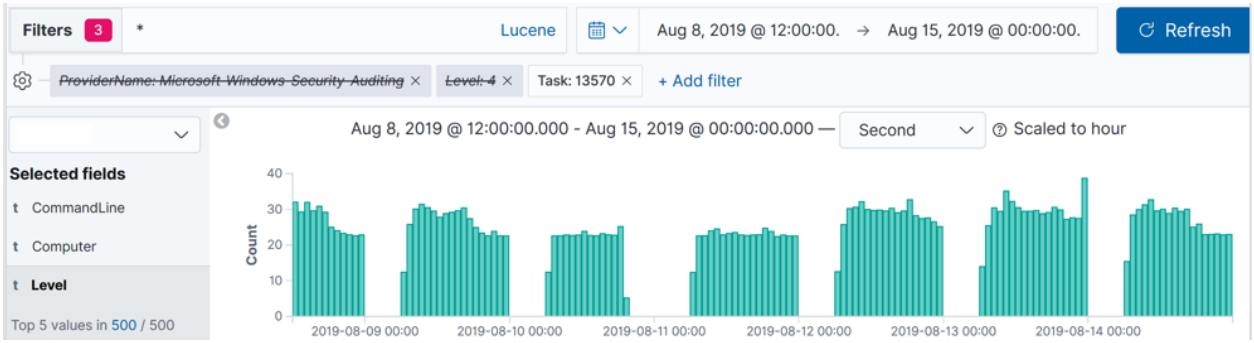


Figure 5-2 Number of events from Level 4 over a one week period.

- All events occur between 06:00 and 23:59
- All computers have a task 13750 at 23:59:59

- This task did not seem unique to the end of the day. It appears that it happens consistently through out the day. See Figure 5-2.
- It looked like there is consistent activity throughout the day apart from the Microsoft-Windows-Activity-Security-Auditing Provider Name (Figure 5-3).
- There seemed to be a periodic nature to the data.
- There appeared to be an interesting activity on Sunday 6/23/2019, that is, t is the only activity on a Sunday.
- Examining the "expected" ordering of the log-ins (EventID 4624) and log-offs (EventID 4634), the ordering was unexpected. 4624 will be called several times without a logout. The same with lock (EventID 4800) and unlock (EventID 4801).
- It did seem as though 4800 and 4801 occur in pairs.
- The pattern for locking seems to be:
  - 4801, 4624, 4624, 4634, 4634, 4800,
  - First 4624 has an ElevatedToken: Yes, and
  - Computer: USER000032 (anonymized user) has a pattern that only has one 4624 and 4634 rather than two.

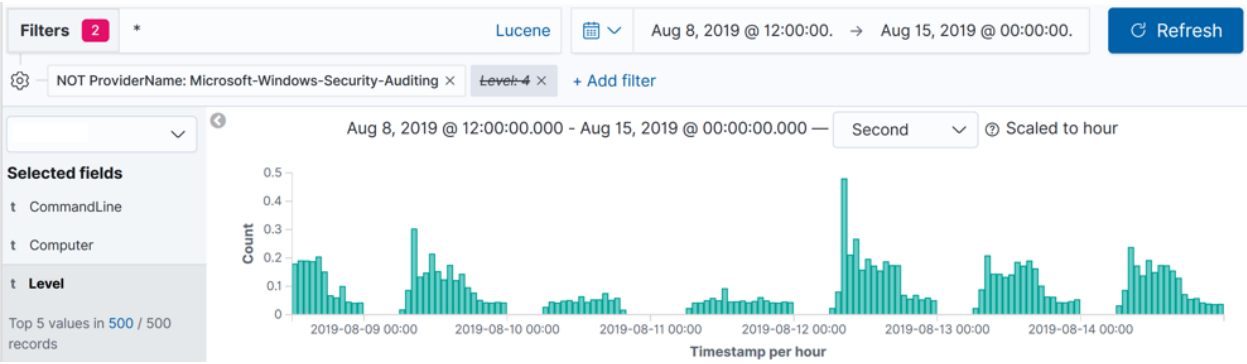


Figure 5-3 Number of events from Task 13570 over a one week period.

5.3.4. Modeling Normalcy

In this section, we seek to understand what constitutes "normal" activity such that it can be modeled. We expect that any significant deviation from normal can be flagged as anomalies. There are several approaches to determine what is normal. In this section, we focus on the number of logs and drill down, building off of the previous findings, to filter out a lot of the noise that comes from standard background processes to hone in on anomalous user activity. We will examine the number of logs per day and per user. We found that there is a strong periodic pattern and that pattern varies considerably for each user. The pattern was most noticeable when filtering

out the events with the "Microsoft-Windows-Security-Auditing" ProviderName as those logs overwhelmed all of the other logs.

As expected, we observed an increase in logs as the number of users increases. The number of logs was fairly consistent the last 4-5 weeks of the data gathering period. Any peaks or dips in this number would indicate an anomaly, such as a system outage, and would be beneficial from a systems operations point of view. Examining the number of logs on a daily basis, we could observe this as shown in Figure 5-4. The dip on 7/6/2019 shows indicates a possible system outage (in fact, there is no data for that date). Aggregating at the week-level provides similar information, however, a single day with missing data could be lost.

Figure 5-4 Sample total number of logs per day.

As expected, User activity varies considerably among users. The magnitude of the counts, however, was quite surprising. Figure 5-5 shows the daily number of logs from users USER000001, USER000002, and USER000003. Each user has a unique pattern for their number of events logged per day. User USER000001 has a very stable pattern during the first 4-5 weeks of the data gathering period with a long dip (a possible vacation). User USER000002 had more sporadic behavior and does not finish the data gathering period. Here, did something go wrong with the software? Was this expected? User USER000003 had significantly lower number of logs during the week with more variability that user USER000001. During the weekend, this user did no work compared to USER000001 who still has activity even on the weekend.

The data shows that logs from "Microsoft-Windows-Security-Auditing" processes dominates the logs from the other processes. Our hypothesis is that the logs from non-"Microsoft-Windows-Security-Auditing" processes will reveal more activity that a user initiates. In the current data set, all of the logs have a level of 0 referring to the lowest security risk.

Examining user USER000003, the logs from the "Microsoft-Windows-Security-Auditing" processes had a different behavior from the non-"Microsoft-Windows-Security-Auditing" processes. Figure 5-6 shows the difference in magnitude and behavior between the "Microsoft-Windows-Security-Auditing" and non-"Microsoft-Windows-Security-Auditing" processes. Removing the logs from the "Microsoft-Windows-Security-Auditing" processes

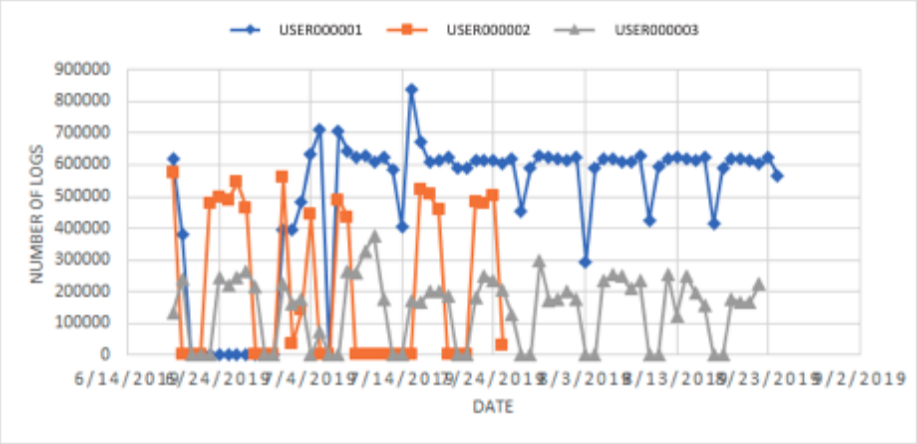
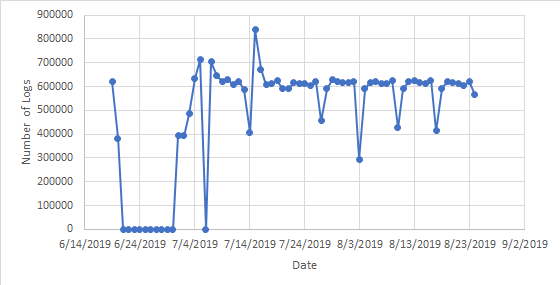
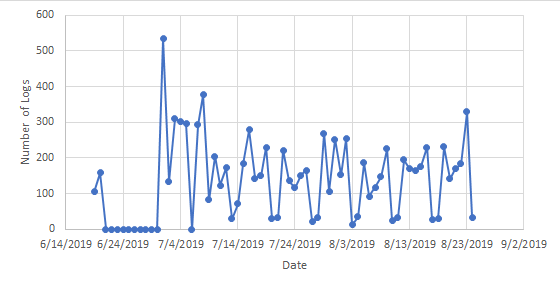


Figure 5-5 Total number of logs per day, compared across three users.

reduces the amount of logs to the order of 100's of logs rather than 100,000's. The periodic behavior was also different. In the "Microsoft-Windows-Security-Auditing" log sources, the number of logs on each week day was fairly constant. There is a spike on 7/15/2019 that could be investigated, however, as this user gets more established, the data is fairly constant.



"Microsoft-Windows-Security-Auditing" processes



non-"Microsoft-Windows-Security-Auditing" processes

Figure 5-6 Comparison of the number of logs from "Microsoft-Windows-Security-Auditing" processes and non-"Microsoft-Windows-Security-Auditing" processes.

For the non-"Microsoft-Windows-Security-Auditing" processes, the number of logs varies significantly across users. For this user, there seemed to be spikes on Mondays and Fridays. Removing the "Microsoft-Windows-Security-Auditing" logs reveals this behavior. This initial analysis suggests that there are different behaviors based on the type of logs that ar generated. We drilled down on the ProviderName field, however, more in depth analysis on other fields or other values for the ProviderName could yield other interesting insights.

Finally, to give a more global view of multiple users, Figure 5-7 shows the number of logs from non-"Microsoft-Windows-Security-Auditing" processes for ten randomly selected users. This figure shows that the users' behaviors can vary quite significantly in terms of the number of logs logged and the overall behavior. User USER000018 has more logged events than the other users

most days of the week. However, many of the other users have very similar behaviors and similar magnitudes of logged events.

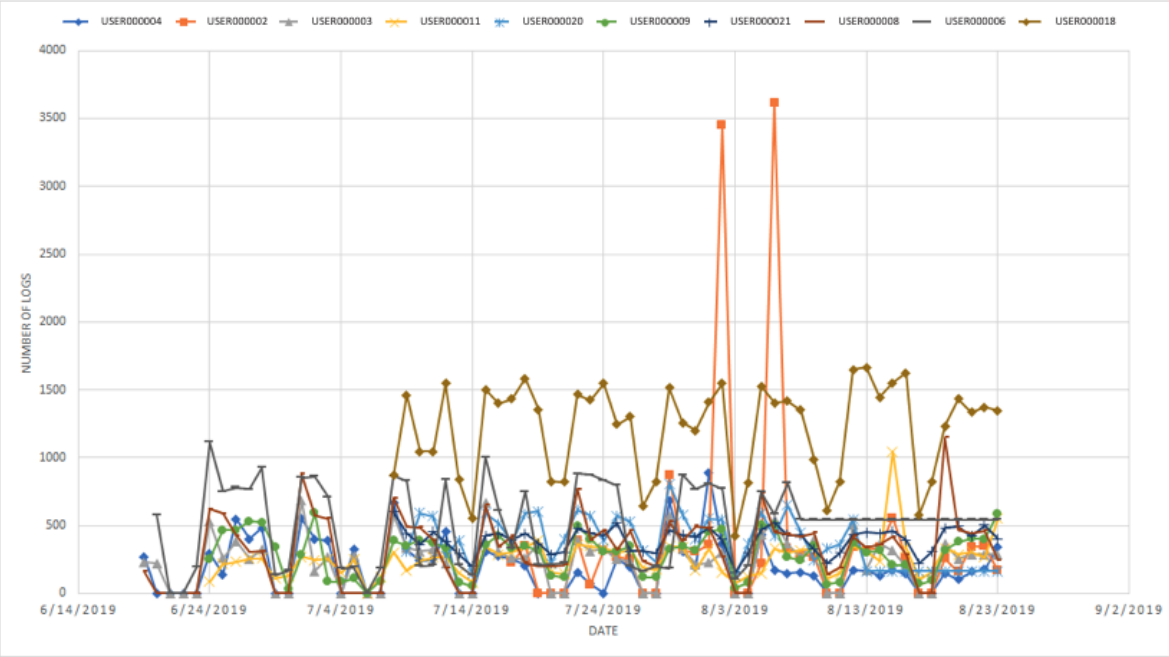


Figure 5-7 Total number of logs per day from non-"Microsoft-Windows-Security-Auditing" processes for 10 randomly selected users.

The behavior from user USER000002 had two very noticeable peaks in its data, where most of the other days align roughly with the other nine users. These two days are obvious outliers and should be investigated further.

The discussion we provide here, shows that there is strong regularity within the logged data. However, due to the large amount of data, some down sampling can help improve that signal and reveal different aspects of the data. Future work should include periodic modeling of the data and fitting distributions over the data allowing for some model of normalcy.

5.4. Team Three

We split on the first day to explore the data more broadly and dive deeply into the EventID field. See Section 6 for the extension from exploratory data analysis into the analytics.

5.4.1. Feature Relevance

For the first day we wanted to get an understanding of the data. This would allow us to filter our high level view in order to find interesting things in the data. The data is very scarce because most fields only occur with a given event ID. In total there are 822 unique fields in the data. It would be

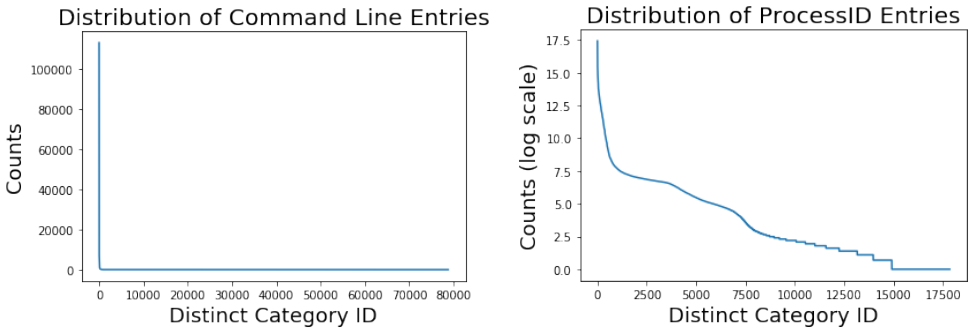


Figure 5-8 Distribution of Distinct Categories in Command Line and Process ID Fields

prohibitively time consuming to label the relevance of each field in the few days of the workshop. Out of the 822 unique fields there were 331 fields that only had one distinct value in them, so the field occurring only acted as a flag. There were 116 fields with only two categories. This meant about half our fields held limited additional information once we knew they existed. The fields that contained greater than 5000 categories were: CallerProcessId, CommandLine, CorrelationActivityID, Data, Data1, EventRecordID1, EventSourceId, ExecutionThreadID1, FilterId, FilterKey, HandleId, IpPort, LogonGuid, NewProcessId, RuleId, TargetLinkedLogonId, TargetLogonGuid, WindowStation.

Some of the categorical fields make sense and others were not investigated in the time available. Process ID makes sense to have many values occurring because the system randomly assigns them from the available range. However, on the right plot in Figure 5-8, we show the distribution of the different process IDs and see a strong skew. If each process was driving similar number of events and process IDs were generated randomly we would expect a more uniform distribution. On the left plot on the same figure we see that almost every observation of our Command Line captures were single observations. Some of this is due to full paths being captured in command calls, some command lines being system executed so specific process IDs were connected with those calls. A direction we did not investigate was to remove the path from all the calls to clean them and then look at using fuzzy matching to cluster similar calls.

There were only 9 fields which occur in every record, (Channel, Computer, EventID, EventRecordID, Keywords, Level, ProviderName, Task, and Timestamp). Our interest in day two turned towards these fields to explore what we could discover without the problem of data scarcity.

5.4.2. EventID Analysis

On the first day, we mainly focused on looking at basic statistics of the EventID distributions. Figures 5-9 and 5-10 show the mean EventID counts and their distributions, respectively, for the 33 different users in our dataset. EventIDs have been reordered by counts, in descending order. The most common event occurs over 36 million times, on average, while ~ 100 events have mean counts greater than 10, and ~ 200 events have mean counts greater than 1. The distribution of event IDs is highly skewed.

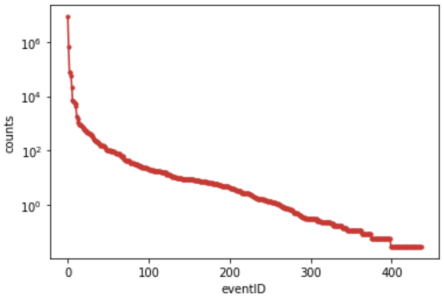


Figure 5-9 Mean counts of the 437 unique events, averaged over the 33 different users in our dataset. The IDs have been reordered by counts, in descending order.

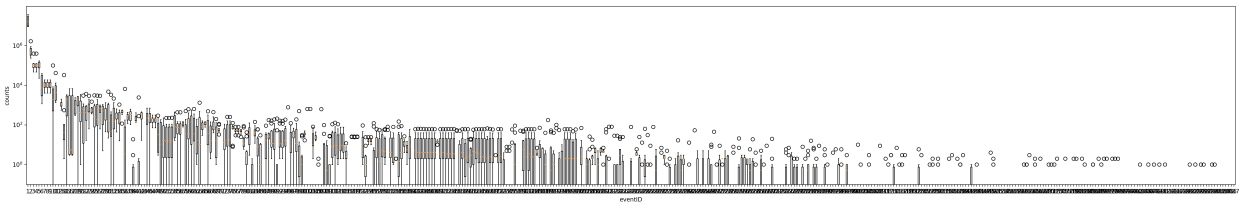


Figure 5-10 Box plots of the 437 unique events, showing the distribution over the 33 different users in our dataset. The IDs have been reordered by counts, in descending order.

5.5. Team Four

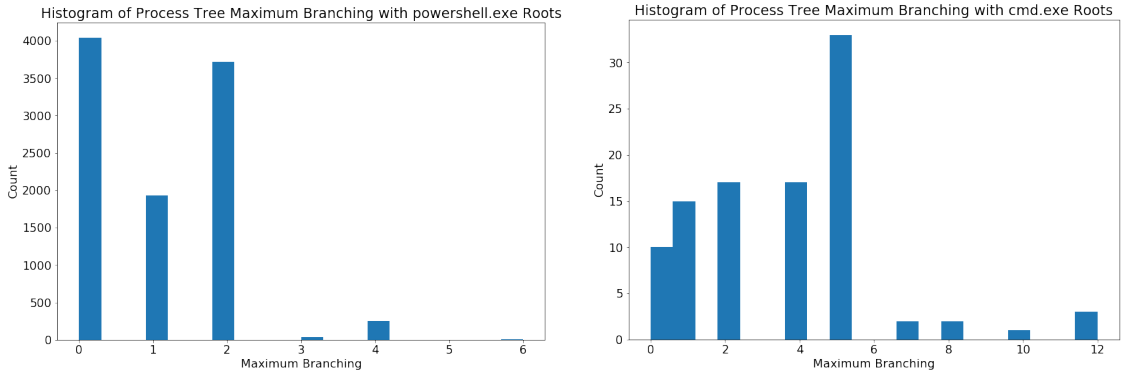
In our fourth goal for the workshop, we wanted to explore process chaining, beginning with more interesting processes from a cyber defense viewpoint. For example, Windows powershell execution is often used for both staged infection and for lateral movement. Thus, we explored the statistics of process trees containing powershell execution, which turned out to be more interesting. Consequently, this exploration was entirely focused on Windows Event ID 4688 (process creation). During the workshop, our exploration was focused on the statistics for process trees containing powershell and cmd only (Windows native or batch command execution).

5.5.1. Methodology

The following steps were performed to analyze process trees:

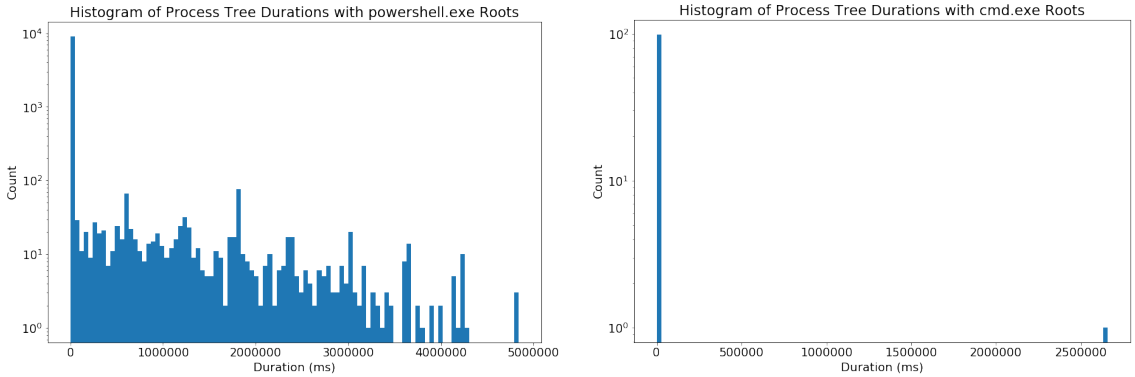
1. Query Elasticsearch data set for processes of interest (i.e., powershell or cmd). Set each result to the root of a tree.

2. Recursively query Elasticsearch for all child processes up to a limited depth. For this analysis the maximum depth was set to 5, which was more than enough given the results below. Assemble these children into a tree structure.



(a) Powershell **(b)** cmd

Figure 5-11 Maximum Branching Factor for Process Trees with a Powershell Root



(a) Powershell **(b)** cmd

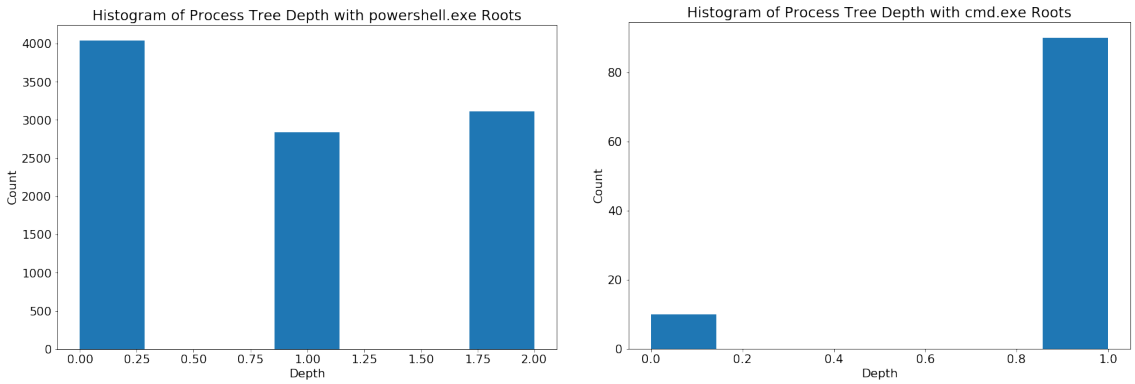
Figure 5-12 Process Tree Duration

3. Measure various quantities about the assembled trees, including maximum branching, maximum depth, and inter-process timing.
4. Record the parent process of the root to condition some probabilities.

Mostly, this methodology was entirely exploratory. However, the focus was on inter-process (parent-to-child) timing under the hypothesis that human-based interaction will be 1) of higher variance, and 2) of higher mean as compared to machine-driven interaction.

5.5.2. Results: Exploration and Characterization

Before measuring timing or other characteristics of process trees, it was important to characterize the structure of the trees themselves. Looking at the tree structure informs expectations of what type of data might be available and useful. For example, should most of the process trees turn out to be single leaves, inter-process communication timing might be meaningless. Consequently, the maximum branching factor, depth, and total duration (root to last child timing) of process trees are all measured.



(a) Powershell **(b)** cmd

Figure 5-13 Process Tree Depth

Figure 5-11 shows the maximum branching (maximum number of children for any parent in a given tree) for powershell and cmd roots, respectively. Figure 5-12 shows the histogram of the total duration of process trees for powershell and cmd roots, respectively. Finally, figure 5-13 shows the histogram of the process tree depth for powershell and cmd roots, respectively. Figure 5-12 shows that the total duration of all constructed process trees with cmd as the root, which mostly is approximately 0 or not measurable by the Windows timing used. Figure 5-13 shows that process trees with cmd roots are either also leaves or have a single level of children (i.e., have depth 1). Finally, in the data collected, there was much less cmd execution than powershell, thus limiting the power of measured statistics. Consequently, the focus will be on powershell below.

5.5.3. Inter-Process Timing

To measure inter-process timing, we serialized all nodes in a tree and sorted by process creation time. Then the times between process creations was measured, allowing us to derive the mean and standard deviation per process tree. Finally, an accumulation for all of these measurements across all process trees was aggregated. Figure 5-14 shows the mean and standard deviation of the inter-process timing for all trees. For each statistic, there is a large spike near 0, which indicates a large fraction of processes exit or create another process almost immediately.

Figure 5-15 shows the inter-process timing for powershell process trees that have a parent process of WmiPrvSe. WmiPrvSe is the Windows Management Instrumentation provider, which is a system service.[1] The inter-process timing given this parent accounts for much of the spike around 0 for both the mean and standard deviation in the overall inter-process timing. This service is used for host management functions, and is likely not to be used by users. Consequently, using this parent node might be useful in labeling data for a supervised classifier. The powershell process trees with parent of kinventory[2] have a similar structure and is likely to also be a good source of

[1] See `https://techcommunity.microsoft.com/t5/Ask-The-Performance-Team/WMI-High-Memory-Usage-by-WMI-Service-or-Wmiprvse-exe/ba-p/375491` (accessed 9/17/19).

[2] A Dell management agent.

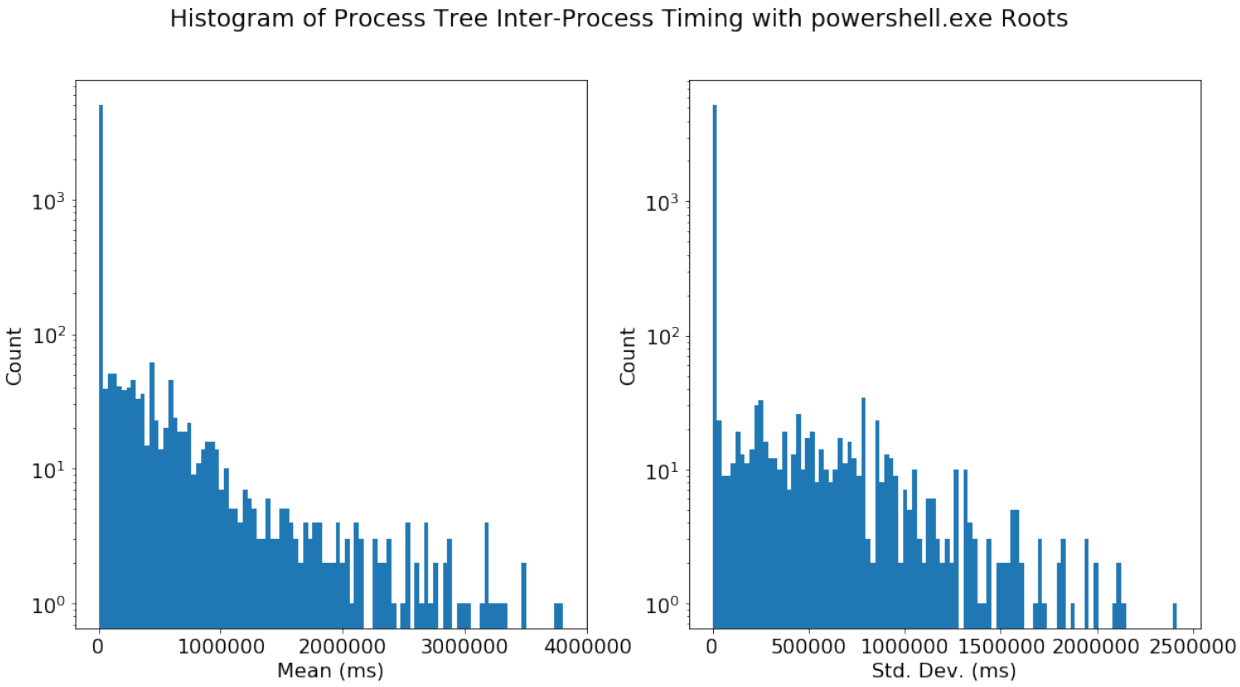


Figure 5-14 Overall Inter-Process Timing for Process Trees with Powershell Root

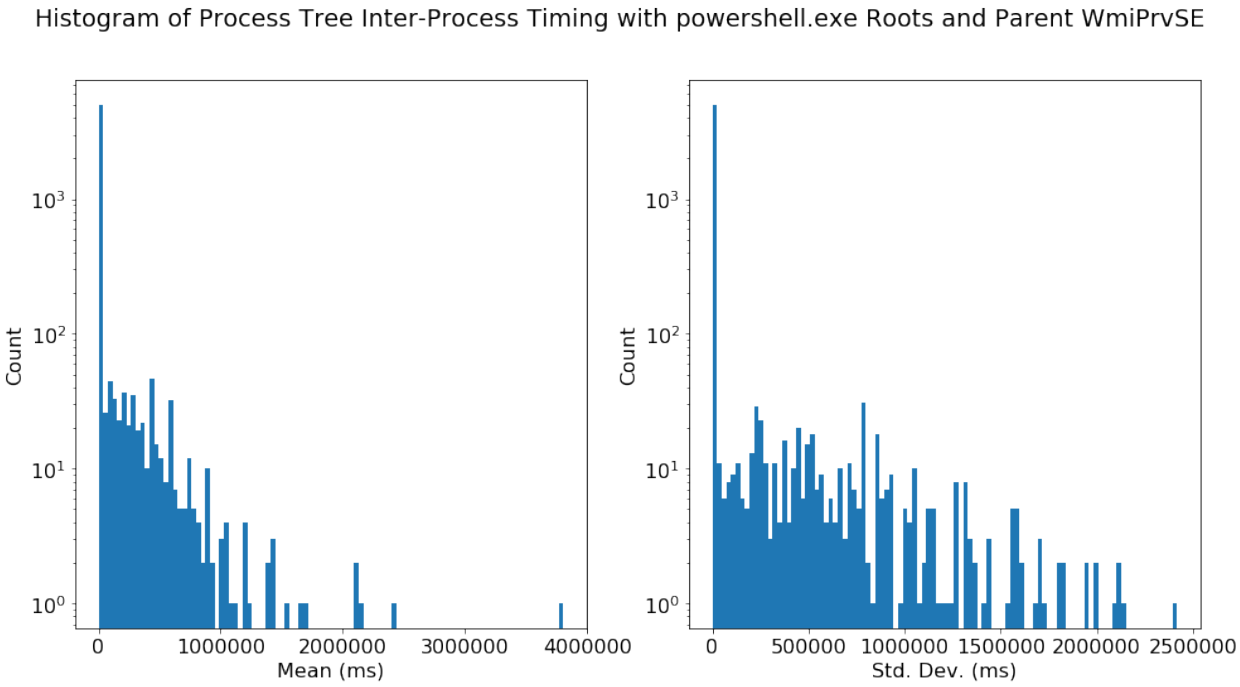


Figure 5-15 Overall Inter-Process Timing for Process Trees with Powershell Root and Parent WmiPrvSE

labeling.

6. PRELIMINARY ANALYTICS

6.1. Brainstorming Activity

Once the teams had a reasonable understanding for the structure of the data and what was in the data (at a high-level). The goal of this project was to establish baseline definitions of user behavior by applying statistical methods to the host based data we gathered. Since the data was so complex, exploratory data analysis took up most of the time. We saw value in leveraging the insights we gained from the data to discuss potential analytics that could be developed in future work and we capture those in this section.

- Understanding and modeling the baseline computer actions without human interaction
- How do we capture and identify outliers in data with extreme sparsity and scarcity?
- Can we classify human driven actions versus automated machine actions?
- With time series analysis are we able to filter out the scheduled behaviors?
- What is an appropriate way to represent users based on logged data? Can we extract user sessions based on power and log on/off events?
- Is there a timing component to logged system events where inter-arrival times of logged events are meaningful?
- The ground truth for holidays exists and it is fairly apparent when computers are powered down for extended times. With that information, can we observe changes in behavior that suggests an upcoming out of office?
- How do we generate process trees based on sequential ordering of events occurring and their interrelated nature?
- Is there a method to ensure temporal alignment of all the logs we receive?

6.2. Preliminary Exploration

Our goal was to categorize normal behavior. Without an understanding of the existing machines and their baseline it is difficult to detect anomalies. In collaboration with our psychology POC we discussed the hypothesis that impulsivity scores may correlate with risky users. This left a few follow on questions. Two related questions were, are users more similar to themselves than others on a time basis, and would a model for each user specifically or all users generally be more appropriate? An initial proof of concept approach to these questions was to consider only counts of EventIDs binned by day. This would give a rough vector representation to how a given user used their computer that day. A user would then have vectors numbering the amount of days they participated in the data collection.

Once we had that representation we used principal component analysis (PCA) to reduce our User-Day vectors to two dimensions where we could plot and visualize relationships, see Fig.

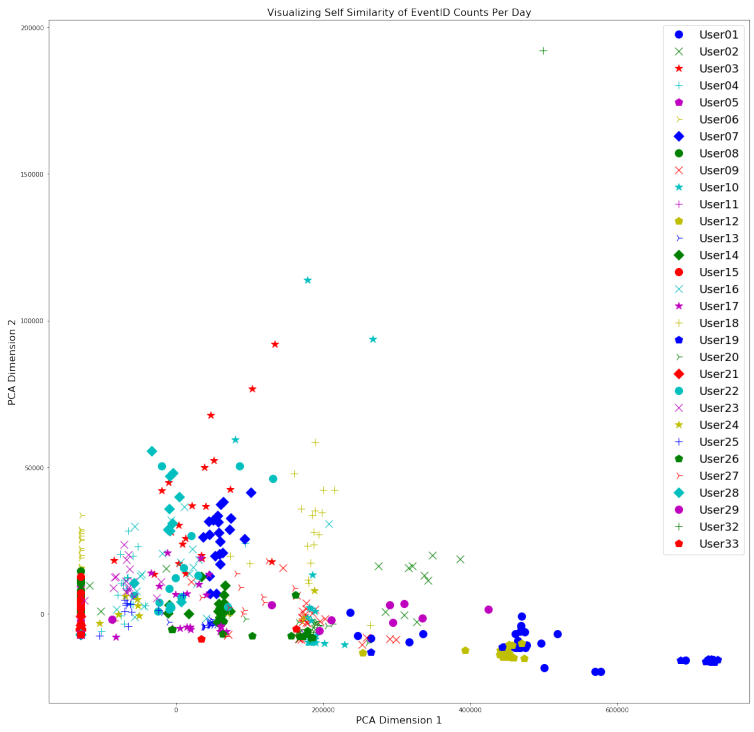


Figure 6-1 Dimension reduction to 2D for All Users

6-1. We include a subset of the users in Fig. 6-2 to allow for easier observation of self similarity for a few users. These preliminary plots suggest that users are fairly self similar from day to day but not so distinct from others that this approach would work to classify our users, based on clustering. As an alternate simple view, we can find the mean of each user and the radius that encompasses all of their reduced dimension points. We can also find the radius of the full data. Comparing the ratios allows us to see most users fall within a relatively small subset of the total observed space. The average ratios across all users is 12%. This metric is skewed by outliers and future work would seek a more appropriate measure.

A second way to represent user daily vectors is through multidimensional scaling (MDS); see Fig. 6-3. This visualization supports the PCA analysis that certain users are self similar from day to day but not so distinct from each other that an approach based on clustering would be able to distinguish users. An exception is the user whose daily EventID behavior is indicated by the blue X markers. Figure 6-4 provides an example of a user that is “normal” (left) and a user that is self-similar, but clearly anomalous with respect to other users (right). It can be seen that the anomalous user high counts for several of the more rare EventIDs.

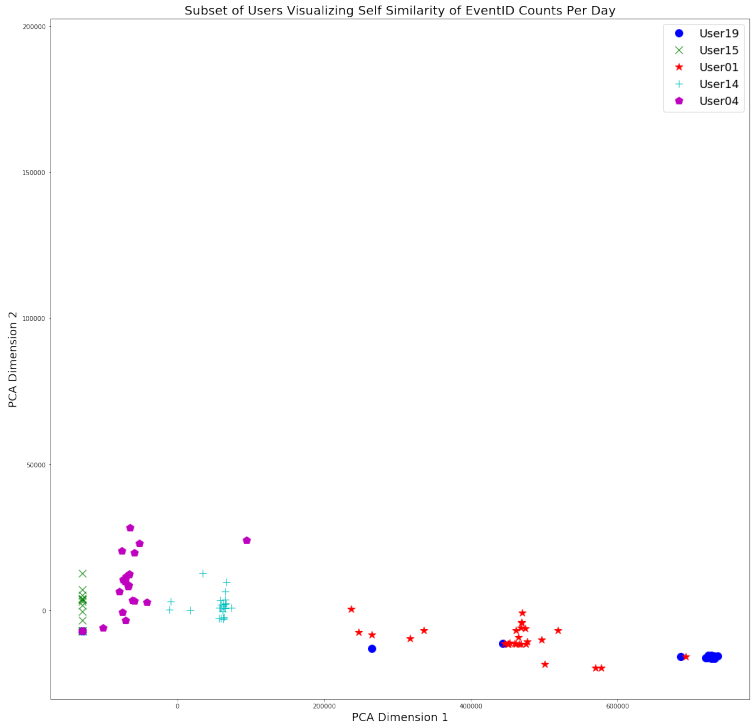


Figure 6-2 Dimension reduction to 2D for a subset of Users

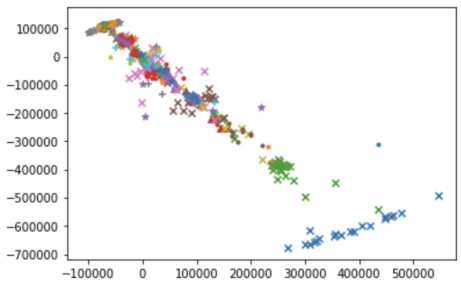


Figure 6-3 MDS embedding of the daily EventID vectors, where color/shape correspond to different users.

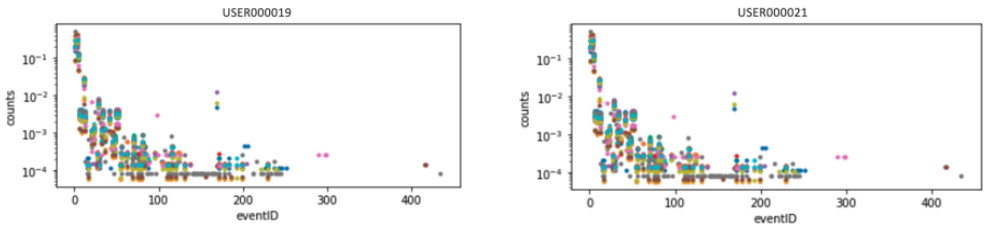


Figure 6-4 An example of a user with a daily EventID distribution that does not stand out as anomalous (left) and a user that does (right). Different colors correspond to different daily event counts of the given user.

7. FUTURE WORK

7.1. Building on what we started

In section 5.5, the work we provided is an initial exploration into the statistics of process trees and requires more investigation. With more time, we would build out parent processes based on interesting root nodes (e.g., the powershell and cmd use above). While powershell and cmd are used for interesting behavior like lateral movement and multi-stage infections, those tools are also used for benign purposes, e.g., computer management. Thus, reducing the search space and number of process trees built by focusing on combining other indicators besides the given executables above would both reduce computation and focus on more interesting behaviors.

Some of the above results were likely from automated processes. We could also search for more manual or direct user interactions to find trees that could be labeled as being user-driven. Potentially focusing on more user-oriented use of the above tools would help lead to that labeling. Having such labeling for both classes, we could then proceed to attempt to differentiate the two, either with the inter-process timing or another metric, e.g., tree depth, number of processes launched, etc.

7.2. Cognitive Psychology

Future work will involve leveraging the preliminary data analysis and results from this project to explore the link between host-based data and the human factor. We intend to expand on the impulsiveness research given that user host-based data is available. We propose to correlate the user's impulsiveness ratings with a risky cybersecurity behavior that is available in the data, such as neglecting to lock one's computer screen. We hypothesize that those with higher impulsiveness scores will be more likely to neglect to lock their computer screen.

Additionally, we propose leveraging well-validated theories and findings from cognitive psychology to make predictions about a user's behavior. For example, it is well cited that humans tend to make errors when their routine changes; we can experimentally test how changes in routine might affect a user's cybersecurity performance. A user might be more likely to make a mistake after a long vacation, for instance, and we can empirically test whether we can identify when a user has taken a vacation based on his/her host-based data.

Another potential avenue is to take what we know about human attention and formulate a hypothesis about how attention will correlate with users' cybersecurity performance. For instance, it is well known that humans can sustain attention for a short period of time; can we identify the difference between a human and a computer in the host-based data based on some measure of attention (such as taking breaks or task switching)? We also know that humans tend to make mistakes when they are distracted; are we able to identify when users are distracted and potentially suggest recommendations for minimizing the likelihood of mistakes? These are empirical questions that can be tested experimentally. Continued efforts to understand and parse the collected data will allow for these ideas to be further refined and improved.

7.3. Research Direction: Anomaly detection

We will use the preliminary research resources (datasets) obtained in this project to facilitate anomaly detection with machine learning or deep learning methods to protect operational enterprise networks from intentional cyber attacks or unintentional human errors. Our goals of the anomaly detection are (1) detecting abnormal host behavior caused by cyber attacks that bypass network security (e.g., malware), and (2) identifying unauthorized users physically accessing the hosts or legitimate users making human errors (e.g., legitimate credential is stolen or legitimate users opening malicious attachments).

7.3.1. Modeling user-behavior at the event level

In order to build models of normal behaviors, we need to identify appropriate representations of the events incurred by a user on a particular periodicity. Behaviors may be "seasonal" because a user may repeat their behaviors on a daily basis. We would seek one or multiple time-dependent dynamic models to ground our understanding of behaviors.

7.3.2. Correlate between human impulsivity and event-level dynamic models

Correlating human impulsivity and event-level dynamic models allows us to characterize whether users' event-level dynamic behaviors are correlated to their impulsiveness. Insights into this matter will allow us to understand how human factors may affect their behaviors. This understanding would guide us to construct robust anomaly detectors that can account for variance in human behaviors.

7.3.3. Generating abnormal host-based data for malware analysis

In our research efforts we obtained ambient host-based data to help understand normal behavior of an operational enterprise system. A next step is to define an attack model and collect host logs when the computer is compromised by such cyber attacks. For example, we could collect malware samples and infect hosts with malware in a controlled environment. The abnormal host-based data could be used to develop dynamic malware analysis tools to identify the true intent and capabilities of malware. The infected host data would also provide a series of technical indicators for malware early detection and protection before catastrophic malware could infect the whole enterprise network.

REFERENCES

[1] Open source search and analytics. https://www.elastic.co/.

[2] Windows Logging Service. https://www.federallabs.org/successes/success-stories/windows-logging-service.

[3] Sherly Abraham and InduShobha Chengalur-Smith. An overview of social engineering malware: Trends, tactics, and implications. *Technology in Society*, 32(3):183 – 196, 2010.

[4] Kansas City National Security Campus. Windows logging service. https://kcnsc.doe.gov/docs/default-source/kcnsc-software/windows-logging-service-summary_073117.pdf?sfvrsn=26b745c4_2. Accessed: 2017-08.

[5] Nabie Y Conteh and Paul J Schmick. Cybersecurity: risks, vulnerabilities and countermeasures to prevent social engineering attacks. *International Journal of Advanced Computer Research*, 6(23):31, 2016.

[6] Joseph M Hatfield. Social engineering in cybersecurity: The evolution of a concept. *Computers & Security*, 73:102–113, 2018.

[7] Brent A Kaplan, Michael Amlung, Derek D Reed, David P Jarmolowicz, Todd L McKerchar, and Shea M Lemley. Automating scoring of delay discounting for the 21-and 27-item monetary choice questionnaires. *The Behavior Analyst*, 39(2):293–304, 2016.

[8] Ross Kelly. Almost 90% of cyber attacks are caused by human error or behavior. https://chiefexecutive.net/almost-90-cyber-attacks-caused-human-error-behavior/, March 2017.

[9] Kris N Kirby, Nancy M Petry, and Warren K Bickel. Heroin addicts have higher discount rates for delayed rewards than non-drug-using controls. *Journal of Experimental psychology: general*, 128(1):78, 1999.

[10] Carl W Lejuez, Jennifer P Read, Christopher W Kahler, Jerry B Richards, Susan E Ramsey, Gregory L Stuart, David R Strong, and Richard A Brown. Evaluation of a behavioral measure of risk taking: the balloon analogue risk task (bart). *Journal of Experimental Psychology: Applied*, 8(2):75, 2002.

[11] Ralph C Smith. *Uncertainty quantification: theory, implementation, and applications*, volume 12. Siam, 2013.

[12] Emmanuel Trouche, Emmanuel Sander, and Hugo Mercier. Arguments, more than confidence, explain the good performance of reasoning groups. *Journal of Experimental Psychology: General*, 143(5):1958, 2014.

[13] Melissa J. M. Turcotte, Alexander D. Kent, and Curtis Hash. *Unified Host and Network Data Set*, chapter Chapter 1, pages 1–22. World Scientific, nov 2018.

[14] Rishi Vaidya. Cyber security breaches survey 2019. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/791940/Cyber_Security_Breaches_Survey_2019_-_Main_Report.PDF.

DISTRIBUTION

Hardcopy—Internal

Number of Copies	Name	Org.	Mailstop
1	D. Chavez, LDRD Office	1911	0359

Email—Internal (encrypt for OUO)

Name	Org.	Sandia Email Address
Technical Library	01177	libref@sandia.gov

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.