SMU Data Science Review

Volume 1 | Number 4 Article 5

2018

# Project Insight: A Granular Approach to Enterprise Cybersecurity

Sunna Quazi
*Southern Methodist University*, squazi@smu.edu

Adam Baca
*Southern Methodist University*, agbaca@smu.edu

Sam Darsche
sam@felixsecurity.net

# Project Insight: A Granular Approach to Enterprise Cybersecurity

Sunna Quazi[1], Adam Baca[1], and Sam Darsche[2]

[1] Southern Methodist University, Dallas, Texas 75205
[2] Felix Security, Level 32, 101 Miller Street, North Sydney, NSW 2060, Australia
{squazi, agbaca}@smu.edu,
sam@felixsecurity.net

**Abstract.** In this paper, we disambiguate risky activity corporate users are propagating with their software in real time by creating an enterprise security visualization solution for system administrators. The current problem in this domain is the lag in cyber intelligence that inhibits preventative security measure execution. This is partially due to the overemphasis of network activity, which is a nonfinite dataset and is difficult to comprehensively ingest with analytics. We address these concerns by elaborating on the beta of a software called *"Insight"* created by Felix Security. The overall solution leverages endpoint data along with preexisting whitelist/blacklist designations to unambiguously communicate potential Indicators of Compromise (IOC) on an executive level by employing visualizations.

## 1 Introduction

To present clearly is synonymous with presenting in black and white. Enterprise security may have attempted to borrow from this concept by defining Whitelist programs, programs that are essential for operation and verifiably safe; and Blacklist programs, programs that are either known to be malicious or indicative of unsanctioned activity in the workplace [4]. The concern with this approach is that every uncategorized program ends up in a grey area, causing the line between white and black to become more of a nebulous cloud of uncertainty with each node denoting a possible IOC. The purpose of this study is to delve deeply into categorization of these programs by using a software called *"Insight"* to determine any and all possible insights that can be extorted from the dataset it generates to not only satisfy cybersecurity ends, but also general workplace analytics.

The common premise between Whitelists and Blacklists is that you are inventorying programs that are known, even though there is no singular inventory of all the programs that currently exist. Whitelisting and blacklisting are used in all areas of computing, not just software, and when it comes to user Access Control Lists (ACL's), it makes sense to only whitelist a small subse [1]t. However, cross-applying this philosophy to an enterprise's ability to access software can inadvertently make a company less agile and delimit its ability to compete and stay profitable by excluding newer low-cost programs.

The *Insight* program ameliorates the infinitely regressive data issue by taking advantage of the fact that the number of programs that run on a machine is finite. Specifically, *Insight* is a program that analyzes how much compute resources are consumed by an endpoint over time. This visualizes the peaks and troughs for workloads processed by the endpoint. The visualization of this data helps Business Users and System Administrators understand the amount of resources (excessive / sufficient / insufficient) which has been allocated to the workloads on the physical or virtual machine. By utilizing the *Insight* program, we access and log the process name, host name, owner name, and IP address every handful of seconds that a computer is running. Amassing a dataset that contains each program that runs on a machine, multiplied by the number of endpoints that exist in even a small corporation could quickly become caustic and overwhelming. Prior to aggregation, one month's of raw data for a single user can easily result in 18 million records totaling almost 2 gigabytes in size. These transactional runtime logs that we are leveraging have always existed on machines, since the advent of the computing age, but have remained underutilized. If Big Data is a problem, this is the Original Sin.

We did not approach this solution as a way of creating a conclusive index of what is good and bad, but rather leveraging preexisting lists that a corporation has already curated and tailored to their specific needs and then extracting the most promising gauges of the cyber health of an enterprise. This solution doesn't sell complexity, which is what you already have with that dense unstructured dataset that was too nauseating to ever think about diving into. *Insight* sells simplicity, and more importantly accessibility. Our first solution is the use of visualization to emote the dataset and inspire action only when it is necessary and provide assurances when everything is just fine. The mentality behind this can be synonymized with soaking rice you had in storage. Only the grains that have been compromised by weevils float to the top to be addressed and skimmed while all the acceptable kernels rest unfettered at the bottom of the pan.

Subsequently, we append the utility of this program to go beyond just cybersecurity. IP addresses can be used to determine where an employee is working and if he is performing unauthorized work from an immigration perspective in a geographic location that he does not have a legal work permit in. The metadata and coupling of like processes can be used as a way for an employee to have a solid rebuttal that his vague misunderstood program actually has utility. This would be useful in many companies that are still running a DOS backbone and where the use of new and upcoming program like R is bizarre and foreign in their IT culture. Lastly the time stamp data can be leveraged for workforce planning, to understand if employees are performing work during working hours, if non-exempt employees are performing work during non-working hours, and for picking apart whitelist data to see if what an actual utilization of an employee is by each specific job role, since job roles in technical environments are often identified by programs that are used.

Our result is an amended version of our *Insight* beta which created five visualization sub-products that were affectionately named based on their appearance for easy reference. A visual walkthrough is presented at the end of this study, but to itemize them briefly: the first diagram allows the system administrator to get to know their users, by their program consumption. The second shows the admin the entire enterprise ecosystem by what their programs are classified as. The third diagram negotiates liability between the first and second visualization by ranking which users create the most risk. The last two diagrams are workforce planning related as they analyze users based on the time and locations they are working in. Our conclusions do not state a result specific to the data that was used to create the visualization, but rather create a tool in which a security professional can quickly get an overview of their endpoint ecosystem, provide a symptom checker for risks in real time, and provide a medium to discover their own conclusions.

The course of this study first defines what the Insight program entailed in its beta form we received it and the intent of the program. Since the program heavily utilizes the designation of software as either whitelist or blacklist, we discuss the history and potential concerns of whitelisting and blacklisting fundamentals. We then emphasize the need for our solution by itemizing the cost of a breach. As always, when profitability and cost saving measures for a corporation are addressed, the ethical ramifications of action must be weighed, which occurs in section 5. To determine the uniqueness of our solution, we discuss preexisting similar work in the sixth section. Sections 7 through 10 breakdown the process of creating our solution in the following order: gathering data, baselining our data, simulation of the endpoint, and the creation of *Insight*'s data pipeline. Section 11 details *Insight*'s resulting portfolio of five visualization to tackle enterprise security and workplace analytics concerns. We then discuss our visualization theory behind our results. There is then an analysis of the *Insight* program, post completion. Lastly we enumerate our conclusions we reached during the finalization of the *Insight* solution.

## 2 The Insight Program

Each workload on a physical or virtual machine is comprised of compute resources. We create metadata for these compute resources and then by analyzing them, it is possble to identify rogue and malicious processes. Using built in data science analytics, Business Users and Security Analysts obtain an insight into "unusual activity" occurring within their workloads. Furthermore, Security Analysts are able to create simple lists of "good and expected activity" and "bad malicious activity". The "good activity" is noted but ignored in the data analytics so as not to taint the data presented and the "bad activity" is identified and ignored so the Security Analyst is aware of its presence and can them prevent it using various security technologies. The remaining data, which is neither 'good" or "bad", are presented as possible indicators of compromise.

The initial version of the software we received, which has thus far been referred to as the "beta", was a program we received from Felix Security and could install on individual machines of our choosing to harvest data. The program came with a predesignated list of whitelist and blacklist programs to model what a generic company would deem as good or bad. Once installed, the program sends a pulse at a short increment of seconds to record every program that was running on the machine at the time, its CPU utilization, the total CPU usage of all programs, the date, the time, the hostname, IP address, the user within the machine, and its black/whitelist designation. With only these eight vectors, we were instantly able see interesting aspects of our own usage by just running summary statistics. For example, we were both in a Master's program and working full-time during the initial installation, and were surprised by the range of unusual hours we were active on our computers. We also noticed that there was a large portion of programs that were not initiated by our own user accounts, but rather some variant of a "system" user. As an attacker, this would be a great place to start hiding malicious programs that would have low promise of discovery without a tool like the *Insight* beta to call attention to them.

However novel these discoveries were, the output from the beta was still a very large table that was not user friendly to navigate with limited statistical experience for a single user, much less an enterprise of endpoints. Also, even with our vast lists of black and white programs, there was still a large portion of programs that were not designated as either, but rather as IOC. *Insight* specifically lists these programs as indicators of compromise to accentuate the infinitesimal effect of a whitelists or a blacklists when a majority of programs that are used are largely unknown.

Our intent for this study was to finish the *Insight* program by automating the wrangling the data from the beta, finding relevant data sets to append it to, then slicing the output into strategic views that could be immediately visually digested and understood by a security advisor to determine if something was awry in their ecosystem. This program is unique because it analyzes data that the enterprise already owns, produces results almost immediately after user activity or incident, and creates a disproportionally large amount of insight compared to how horizontally short the captured data is.

## 3 The History and Evolution White and Blacklists

The vast number of programs that enter the market on a monthly basis, make it difficult to classify programs as either white or black. Whitelisting theory is succinctly defined by comparing it to Apple's application store and how they created an environment where only approved applications are allowed into their marketplace. For example, Apple has made it very difficult for a user to use a ringtone from a non-iTunes website because the iPhone's settings direct the user to iTunes to download ringtones. Physical security also adopts the whitelisting philosophy in that most workplaces that require a magnetic badge to enter the building.

Even though whitelisting philosophies are already well established, some experts like including Marcus Ranum have argued that whitelists will continue to become an increasingly important part of enterprise security as technology evolves [3]. Whitelists are the foundation of most company issued computers as they are set up to only allow one employee to access the computer and certain applications within [2]. Ranum's point hinges on the fact that companies will begin to pay more attention to how their hardware is used. Companies should have a better understanding of the programs and destinations that their employees need to use as a function of their job [3].

For the most part, a blacklist has applications and programs that are known to be malicious after an exploit has been found. Blacklists can also incorporate programs that an enterprise does not deem particularly compromising, but rather just counterproductive to work, like Netflix or computer games. Once discovered, the item will be placed in the blacklist to prevent the operator from accidentally or intentionally accessing it. Blacklisting, therefore, has been used as a reactive approach to security, as the list can only be curated after a harmful impact is experienced.

**Table 1.** Advantages [5]

| $BlacklistAdvantages$ | $WhitelistAdvantages$ |
|---|---|
| Easy to manage | More secure |
| Easy to install | More accurate |
| Can download updates quickly | Minimizes false positives |
| | Can be created at various levels within the enterprise |
| | Easy to customize |

**Table 2.** Disadvantages [5]

| $BlacklistDisadvantages$ | $WhitelistDisadvantages$ |
|---|---|
| Exponential growth | More time to manage |
| Many false positives, potentially denying access | Requires additional time to install |
| Continual updates are required | |
| Hard to switch to whitelisting | |

**Potential Vulnerabilities of White and Blacklist.** Regardless of either or a combination of the protocols has been adopted, a concern is that hackers have become familiar with the logic behind whitelists and blacklists. Listing is not new, and the contents of these lists are far from anonymous or proprietary, as third parties such as Alexa have published whitelists to promote more open source development. A published whitelist in the wrong hands can be like a shopping list to create a trojan program, because the enterprise is essentially declaring which programs are likely to remain un-investigated. Cerber ransomware thrives in this blindspot, as the initial exploit often adopts the alias of a whitelisted program, such as an updater, to use as vehicle to the machine. Cerber then works to encrypt the hosts machine's data and demands money in exchange for access to a data owner's files. Xavier Mertens posted an instructional procedure on how a potential attacker can use published whitelists in order to spoof a DNS to host malicious scripts; target users, subsequently, unknowingly download and infect their organizations after initial infection [10]. In blacklisting, if a software is prematurely blacklisted without proper due diligence, it could clog results by generating of tons false positives, and therefore obstructing a security analyst view from actual threats.

## 4 Cost of a Breach

The cost of a data breach can be substantial depending on the size of the organization, which is why our agile visualization solution has utility. The Ponemon Institute and IBM did a study on the cost of a data breach for a sample of organizations across several countries and the average cost of a data breach was 3.62 million dollars per breach in 2017 [6]. There are several factors that contribute to the total cost, including amount of records per breach, type of industry, and even the country in which the company is located. The cost of a breach is something that must be considered when allocating cost to data security.

The main contributing factors to the total cost of a data breach are more than just the potential loss incurred by the sale of stolen data. The industry that a company resides in is the first thing to consider as a business leader. The health industry leads the market with a cost of $380.00 per breached record where the global average is $141.00 for FY 2017. The financial and services industries follow with a per capita (record) cost of $245.00 and $223.00 respectively [6]. These figures can be attributed to the nature of the businesses listed as each of these have a unique advantage for individuals that seek to compromise consumer data, such as the breadth and detail of personal information that is contained in health records, financial account information from the financial firms, and an opportunistic compromise for quick financial gain from the services industry.

The most common root cause of a data breach is that of a malicious or criminal attack. These kinds of attacks account for about 50% of the sampled breaches for FY 2017. Some of the main contributors to increasing the cost on a per capita basis are third party involvement, extensive cloud migration and compliance failures which together can increase the average cost of each record in breach from $141.00 to $183.40 which is a 23% increase [6].

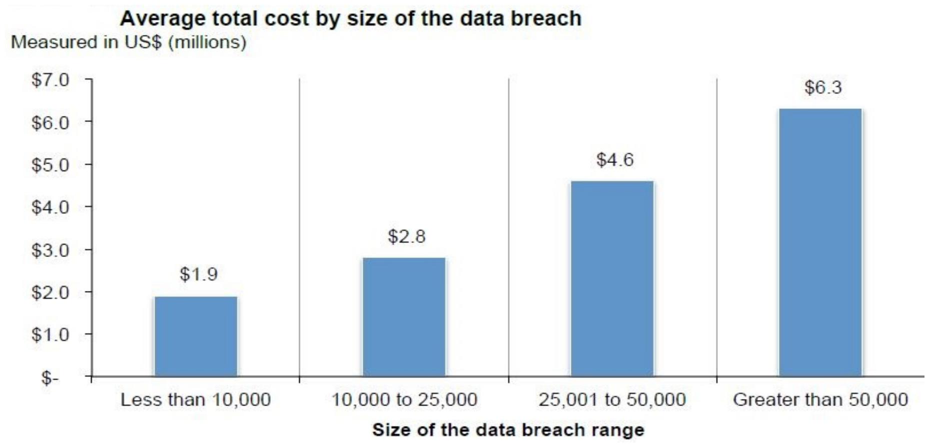

**Fig. 1.** Total Cost by Size of the Breach [6]

## 5 Ethics

Whenever the monitoring of user activity is involved, ethics inevitably comes into question. To respond to the dilemma, we make note that the cost of a breach for a company can easily reach a bankrupting amount depending on the enterprise. Smaller enterprises are particularly vulnerable because the amount of revenue they generate does not always scale with their amount of data and Intellectual Property (IP). Self-preservation is one of the moral criterion that is used to address ethical concerns. If failure to monitor their endpoints results in the end of the enterprise, then they are morally obligated to do so; otherwise they would not exist long enough to have a moral dilemma.

In philosophy, there are a number of doctrines that support this, the first originating as far back as Aristotle and known as "teleology". This can be briefly summarized as: the ends justify the means [16]. Even though this is a moral doctrine, in a contemporary sense this mindset is not highly regarded. However, using "consequentialism," a subset of teleology, we can evaluate the magnitude inaction [15], in this case an enterprise not monitoring it's own data and de facto consenting to a breach, against the means of obtaining data to determine a moral path. As detailed in Section 4, the high cost of a breach, in legal notifications and loss of intellectual property, clearly leans towards action with regards to security monitoring.

The harshness of the above concept is ameliorated when dissecting the moral weight of the "means" in this equation. Most companies have something to the effect of a Use of Company Resources directive, which loosely states that any activity conducted on an enterprise owned device is the property of the enterprise. Onboarding employees generally have to acknowledge that their employer owns the data. Even if it could be argued, that prior to the employee's agreement, the data on the endpoints belonged to each respective employee, US Code affirms that entering into an employer agreement with these stipulations effectually operates as "voluntary disclosure" as it is "lawful consent of the originator" [17].

The solution only monitors activity on devices that an enterprise has knowledge of because they financed them, so the assumption that user activity is being monitored is already there and it is operating on data that the enterprise already owns. *Insight* just merely makes the collection of this activity actually have utility.

Lastly, the solution is not a tool to castigate employees by performing malicious activity because such activity often occurs unknowingly. No single tool, even a cybersecurity one, should be the judge, jury, and executioner on HR related matters. *Insight* seeks to clarify the nature of an endpoint ecosystem and turn information into knowledge. Armed with that advantage, an enterprise can utilize the more contemporary approaches to cybersecurity which involve education, faster reaction, and heightened communication and cooperation [11]. Data has surpassed oil as the world's most valuable resource [14], and it is ethically intuitive to take measures to protect it.

## 6 Relevant Preexisting Work

To ensure that our research had unique ground, we needed to gain a pulse of what preexisting work has occurred on topics in this space. Security Information and Event Management (SIEM) has been around since the early days of the personal computer revolution. Older companies may have implemented a SIEM infrastructure but have failed to keep up with modern threats and capabilities that are currently available to the market. In contrast, some firms see the value in SIEM implementations and are struggling to balance available resources and to meet an effective standard. Factors determining the level of implementation include the companys industry, size, and budget. Because threats and technology are constantly changing, keeping up with millions of event logs and identified threats can become expensive in a short amount of time.

Our concept creates a leaner SIEM inspired tool that generates a comprehensive report at that can still be interpreted on a high level. As we know, the personnel responsible for reviewing IOC's or threat detections must sift through vast amounts of event logs. Daily log records can be in the millions per day depending on the size of the company and it can be overwhelming despite the fact that indicators are a smaller portion of all of the records. Vendors such as ArcSight, LogRythm, Syslog NG, and Splunk have attempted to negotiate these problems while capitalizing on their awareness of challenges facing the industry. Their main offerings are scalability, investigative services and real time threat protection and correlation [8].

Even still, IT leaders are challenged when tasked to report his or her findings to the executive team by showing dense logs that leave the audience with more questions than answers. One of the many focuses of these SIEM tool vendors is that they make event logging and tracking easier and each claim to have a unique algorithm that promises to be superior to their competitors. These are arguably valid directions; however, we prefer a more Occam's razor mentality in our solution; whereas, the complex and obscure algorithm you using to detect your threats are more likely to find a more obscure and less critical threats.

Similar work, in our case, does not only take the form of pre-packaged applications. There are also published datasets on SIEM logs like the Unified Host and Network Dataset found through AZSecure.org, which led to a master data source on the Los Alamos National Laboratorys government website. The data was collected from LANL's enterprise network of Windows machines and the data itself is reminiscent to information that *Insight* provides. The main reason why *Insight*'s data is preferable for this project is that CPU usage is more versatile for use in logging employee productivity as a whole, whereas packet counts are more helpful for logging network focused activity. Because *Insight* provides CPU usage data while the LANL set provides packet counts[9], it is evident that the Los Alamos data set is more network biased while our analysis is more endpoint focused. This LANL dataset is currently the industry model and is heavily saturated within the aforementioned SIEM vendors' scopes of services.

Network data will often show the method of an attack but the actual result that needs to be addressed is entirely encompassed in endpoint data. Even still, network data will not address security incidents created by user error, carelessness, or an inside job. Although the endpoint data we collected is classifiable as "big data", network data is multiplicatively more daunting in mass, which can be attested by any user who has ever run a Wireshark session. After capturing network data, it still needs to be heavily cleaned due to the number of noisy outside influences from anywhere from the internet service provider to conflicting network protocols, designating endpoint data as preferable as a more concentrated source of intelligence.

Finally, there are published guidelines that can be used to comprehensibly build and implement a successful SIEM program. The National Institute of Standards and Technology (NIST) provides guidance on how to create and maintain a secure log management infrastructure [7]. Most of the vendors mentioned above disclose that their implementations are NIST compliant. NIST is a well-known organization for standards and practices especially because NIST compliance is mandatory for federal firms. NIST is a great resource for startup companies that are establishing a cybersecurity infrastructure from the ground up. NIST also acknowledges the limits of log management in that there is a massive amount of data and the difficulty and cost of a highly effective implementation [7]. SIEM can be used to monitor security threats from internal/external actors, policy, productivity, and resource utilization. For these reasons there is quite a bit of research done on the topic, however it is challenging to find research that meets the unique needs of every enterprise. Our *Insight* solution takes advantage of this lack of conclusive research by using the beta to curate our own datasets without preexisting biases.

## 7 Data Gathering

"If I had an hour to solve a problem solution. I would spend the first fifty minutes determining how to... Frame the problem... for once I know the proper question to ask, I can solve the problem in less than five minutes" -an Albert Einstein quote hijacked from a User Experience (UX) advocacy groups twitter feed. This mentality embodies the following section and can help explain why it is so specifically detailed. The utility of *Insight* is equal parts UX with analytics and the way that the raw data is handled initially decided what options are available to us in our solution. Our impression after the data gathering step was that we should be overly cautious and deliberate as to not alienate any possible solutions. Munging is also the period where you become intimately familiar with the dataset, gaining invaluable domain knowledge while tangling your fingers in it to see what bites.

The first challenge was gathering the dummy data we needed to understand how future endpoint data could be aggregated and then visualized. We accomplished this by convenience sampling five different work stations at random intervals ranging from days to months. We locally appended the feeds from our different work stations together to get a base set of commands. These commands were abstracted to automate the data collection process in the future central cloud server implementation. During the munging step, we created an upload vector so we could still access each initial sub data feed directly. We also created the percent CPU vector to quickly denote later on if a specific program was overburdening the system in comparison to its colleagues in the same time interval. Lastly, we casted all vectors, that were initially read in as factors, as dates, times, integers, and character types respectively. Most commands ran relatively quickly until we had to do heavier manipulation of the data, in which we had to employ Graphic Processing Unit ("GPU") computing.

After capturing our full dummy dataset, we parsed it to create three databases. The first just identifies the end user, the time, and their total CPU usage across all programs at that time. This set is used to level set assumptions of what normal CPU usage is, so spikes can be defined as a possible IOC. The second database takes the long data feed initially generated by *Insight* and aggregates it in a wide format. The main difference is that instead of iterating each time a program is alive in a heartbeat fashion, the aggregated database takes each program and gives it a start and an end time. Since the data was aggregated, our previous vectors of program CPU usage, percent CPU usage, and average total CPU usage were averaged. We took specific error handling precautions to make sure that if a program started, stopped, and restarted again, that the programs start and end time were not misrepresented by denoting the first time it ever started for the user and the last time it was ended by the user. This was achieved by creating an instance vector, that was tuned to the intervals the *Insight* program had originally pulsed to query the active programs. Lastly, we created our affectionately labeled golden index database that had every single software instance with their respective stats unmanipulated and every single vector that was created during data munging as reference, spanning thirteen vectors, close to thirty million records, and approximately three gigabytes. It was at this point we realized that the final data warehousing solution for *Insight* would be a cloud based relational database.

## 8 Baselining

In order to demonstrate what is abnormal, we had to run summary statistics off of our current databases. The most accessible method is looking at the total CPU usage of a machine. Below we can determine that the average CPU usage for our sample is approximately .17 with a heavy number of outliers occurring between .6 and 1. We additionally explored this average by determining what was normal by each program. For example, if dwm.exe, which is typically run by the owner Windows Manager/DWM-1 owner is registering as using .4 CPU, when the running average is .003, then that should be considered unusual activity.

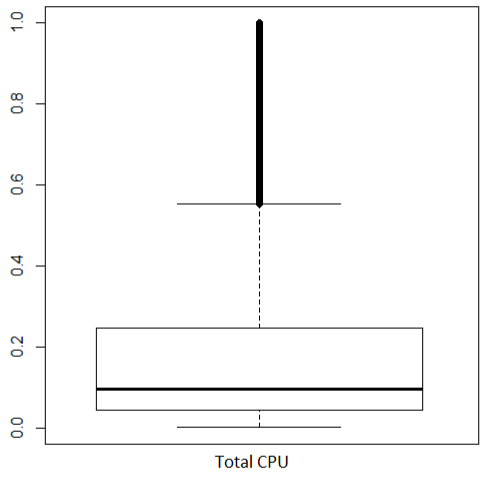

**Fig. 2.** Total CPU Across All Programs and Users

However, these results can be skewed by sleep time as can be seen in Fig. 3. It is normal for a CPU utilization to taper off during periods of inactivity prior to sleeping, so depending how many times a user abandons their computer without shutting down, average CPU usage could be skewed.

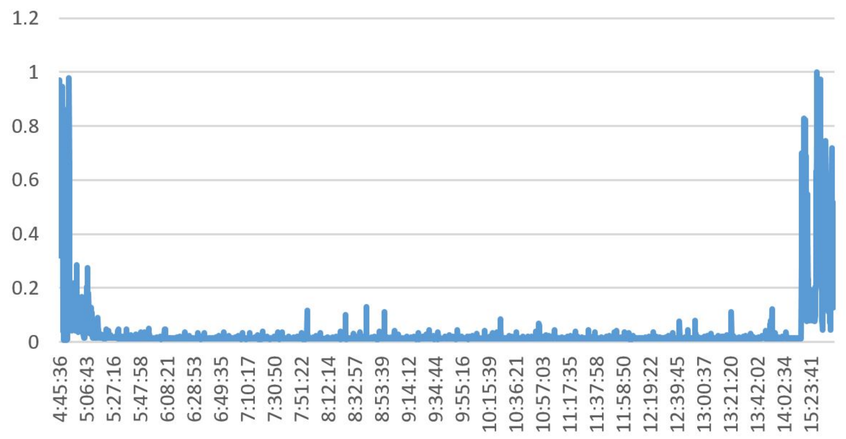

**Fig. 3.** Total CPU Over Time

We tackled this concern by conducting a randomized sleep study to determine the threshold of when a decrease in Total CPU power might indicate user inactivity. We monitored our recorded Total CPU in three different intervals and randomized periods of time to determine what would be the average Total CPU usage during periods of known inactivity and activity (Fig. 4). We used these results to normalize our data so we could decipher our data collection outputs and segregate out inactivity periods of noise. These results can also be repurposed from a workplace planning perspective because it attributes concrete numbers to a period where an employee is perhaps physically present, but not performing computing work.

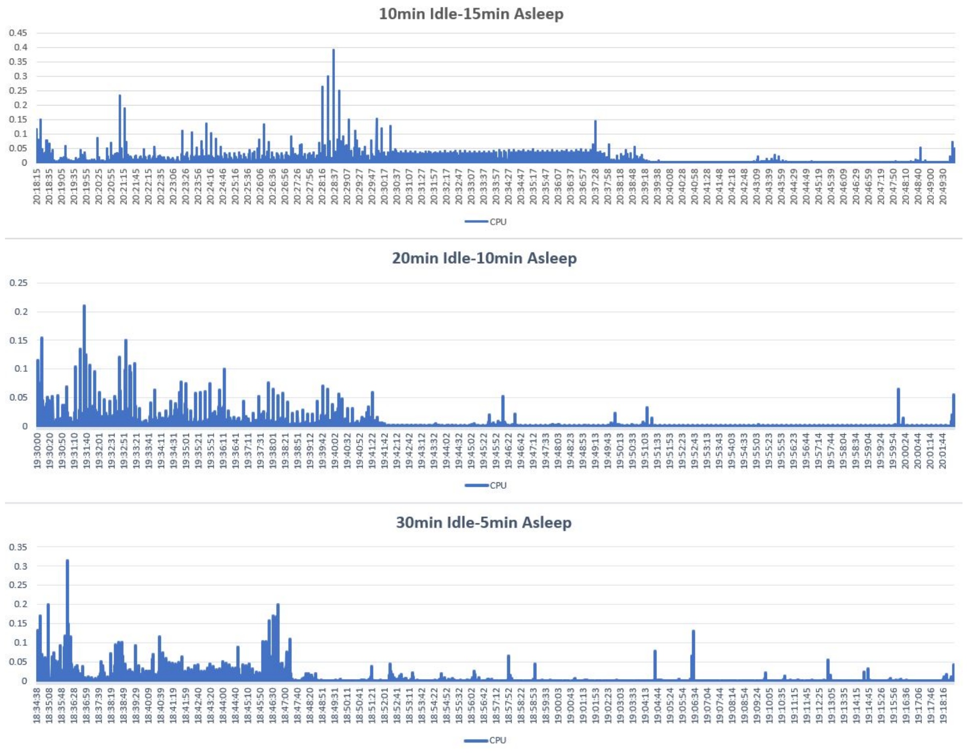

**Fig. 4.** Endpoint Sleep Study Results

## 9 Data Simulation

After we developed our initial data structure we had to engage in heavy sampling to be able to make it more realistic to demo our use cases. The underlying issue is we are trying to model an enterprise environment and we only have five data feeds. We determined that we needed at least 50 endpoints to simulate a small company, with the assumption that all of our visualizations need to be abstracted and scalable for enterprises with thousands of employees, while staying true to our operating criterion that the visualization cannot become too cluttered and still be able to clearly depict possible concerns when needed. Generally sampling is used to create a smaller subset of data that is still representative of a population. We flip that concept on its head by using our smaller data and randomly sampling within it to create more users. To ensure that our resulting simulation data is realistic, we first stratified the data based on the average number of program instances that occurred each day, taking into account the natural variance of data occurring on weekends as opposed to weekdays. The stratification of program instances by weekday vs weekend is our Primary Sampling Unit (PSU).

The average number of program instances run overall was 2231. However, the average for the weekdays was 2431 and the average for the weekends was 1172.

**Table 3.**

| *Monday* | *Tuesday* | *Wednesday* | *Thursday* | *Friday* | *Saturday* | *Sunday* |
|---|---|---|---|---|---|---|
| 3419 | 3379 | 1989 | 3121 | 1642 | 25 | 3518 |
| 6787 | 4581 | 2476 | 4412 | 2662 | 475 | 240 |
| 2485 | 628 | 660 | 3846 | 782 | 1165 | 142 |
| 4018 | 1429 | 1152 | 3325 | 1027 | 2639 | |
| 614 | 1928 | 945 | 1645 | | | |
| 2948 | 2003 | 142 | 5847 | | | |

Secondarily we had to employ a Secondary Sampling Unit (SSU) of clustering by Owner. Before we had collected any data, we assumed that the owner would just be the name of the user we were gathering data from. We soon came to realize that each user generates data belonging to a system owner during their daily activity. To ensure our simulation data was realistic, we ensured that our randomly generated users generated system owner activity at approximate proportions of as indicated below.

**Table 4.**

| *Owner* | *PercentofProgramInstances* |
|---|---|
| Font Driver Host/UMFD-0 | 0.15% |
| Font Driver Host/UMFD-1 | 0.15% |
| NT AUTHORITY/LOCAL SERVICE | 2.17% |
| NT AUTHORITY/NETWORK SERVICE | 1.21% |
| NT AUTHORITY/SYSTEM | 40.77% |
| NT SERVICE/HiveStreamingService | 0.10% |
| Window Manager/DWM-1 | 0.15% |
| USER 1 | 6.65% |
| USER 2 | 14.57% |
| USER 3 | 12.30% |
| USER 4 | 7.09% |
| USER 5 | 14.68% |

## 10 Data Pipeline

Fig. 5 outlines our data pipeline that resulted from Sections 7 through 9 and can be read from left to right. The computer's to the left symbolize the number of endpoint machines. Their individual logs are harvested and then processed and tagged (gears) to ensure that the data remains traceable to the machine for the duration of of the pipeline. The revised data is then collected in a central cloud location. Once there, the data is aggregated and appended to reference tables such as employee rosters or IP to geographic location mapping tables. The data is then manipulated (star) and sliced into custom tables that feed into each visualization. This method allows for whatever language the visualization is written in to not be overburdened with data and have custom recalculated coordinate locations to improve the visualization's performance.

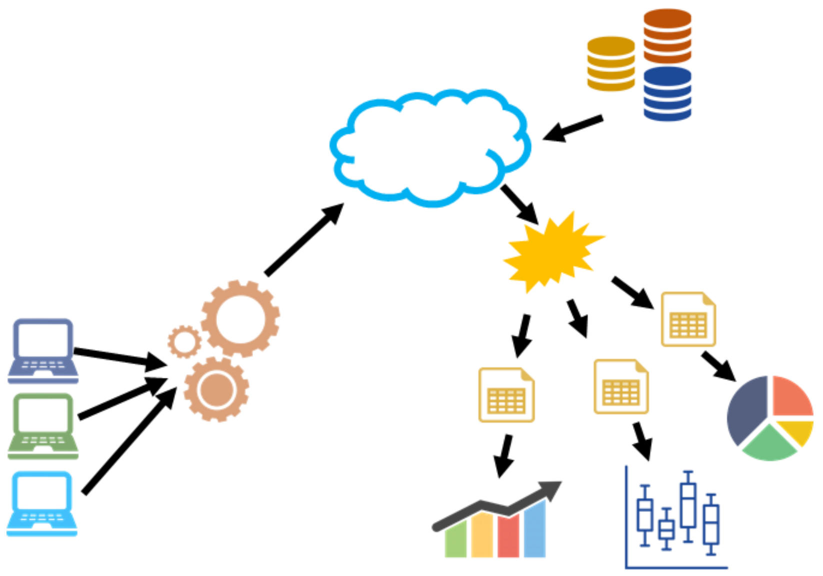

**Fig. 5.** Data Architecture

## 11 Insight Visualization Results

Below are the different visualizations that we wish to showcase to a Security Analyst so they can quickly obtain an understanding of how their healthy their enterprise is performing without being inundated and overwhelmed by the volume of their data. The topics we cover are the three major branches of cybersecurity: Confidentiality, Integrity, and Performance; along with ancillary insights of visualizing Workforce Planning and HR Management. When demoed, these topics are showcased with two visualizations per topic, what the output would look like in a normal situation and what the output would look like in a situation with high IOC or areas of concern.

First, we wanted to give the IT advisor a quick glance at the health of their overall IT environment. The below visualization borrows from our rice metaphor (Fig. 6). Each cluster represents an end point and those with higher blacklist and IOC scores rise to the top. The IT advisor can still see primarily whitelist activity users if they have a curiosity for it by scrolling down, but they are not directed to them initially. If the results of the below diagram are within their predetermined threshold for normal, they can stop investigating after viewing this plot.

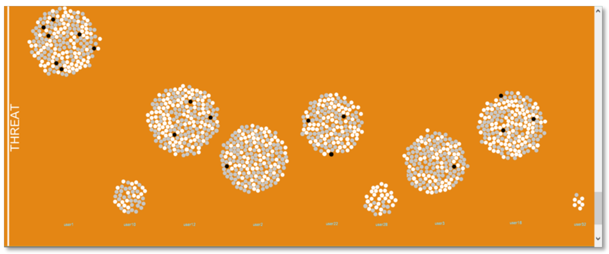

**Fig. 6.** Rice Diagram

Furthermore, after utilizing the vertical dimension, we wanted to give some meaning to the horizontal one. Here we decided to organize our plots by highest CPU usage, with those users who compute more populating on the right side(Fig. 7). This way the security analyst can immediately focus on users with that have the highest threat and are performing the most activity, which would logically spread their threat faster.

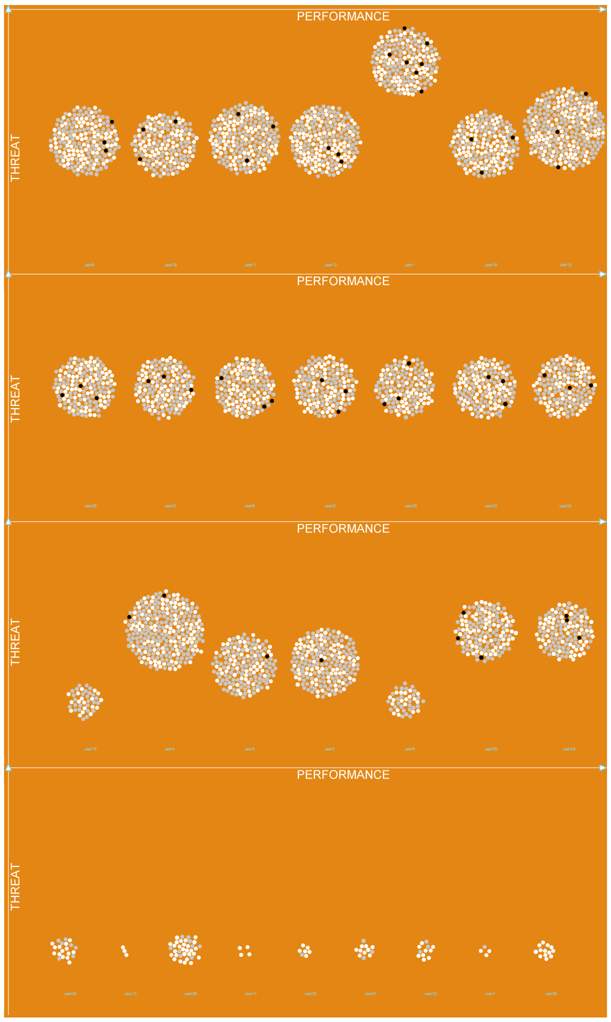

**Fig. 7.** Rice Diagram Full

Secondly, we wanted to allow the IT advisor to see the entire program population of their environment. To accomplish this, we created a more interactive visualization that can be moved and zoomed. This plot borrows from communication visualizations that incorporate the concept of nodes and edges (Fig. 8). The node is either a whitelist, blacklist, IOC program, and the edges would be each individual program in that category. The result naturally separates the programs into different clusters denoted clearly by color. We labeled it "Sinew" because each node can be stretched and moved in an organic motion.

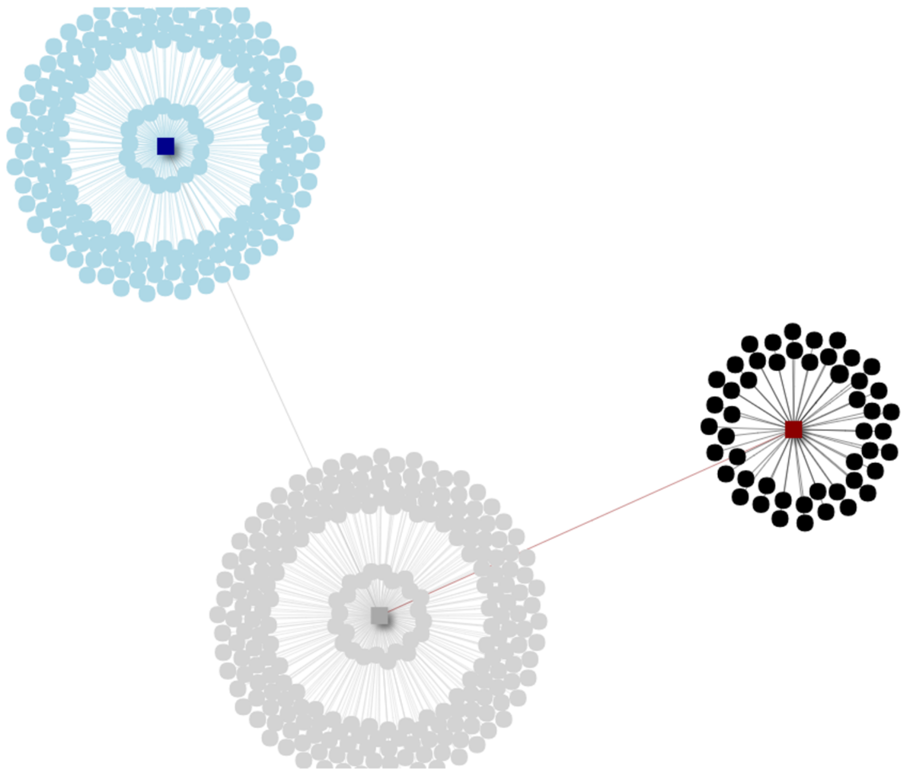

**Fig. 8.** Sinew Diagram

Zooming into a particular node, as Fig. 9 depicts, will give you a breakdown of the programs in that category. Edges that are closer to the nexus are placed in that manner because they have higher CPU utilization and are therefore more relevant. This strategically places more relevant programs closer to the nexus to be noticed as they can be also investigated for performance concerns along with their security categorization.

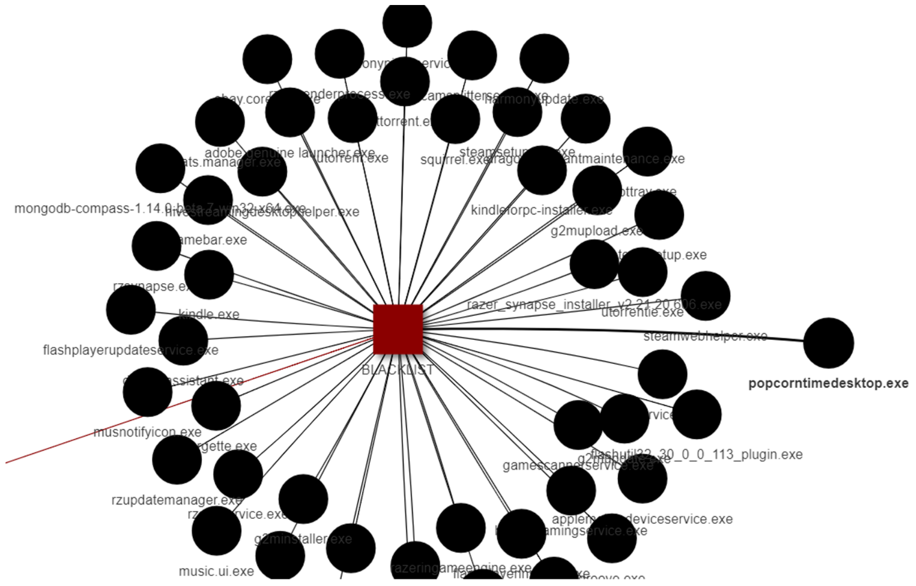

**Fig. 9.** Sinew Zoomed

Another method of employed the community network graph is by our visualization affectionately labeled Liable(Fig. 10). This graph seeks to create a link between Rice and Sinew so you know which users are culpable or not of creating malicious activity. There is a dropdown where you can search for either Blacklist, Whitelist, or IOC types of programs and see each user sized by the time they spend using that category of program to create a sense of accountability in your IT Environment.

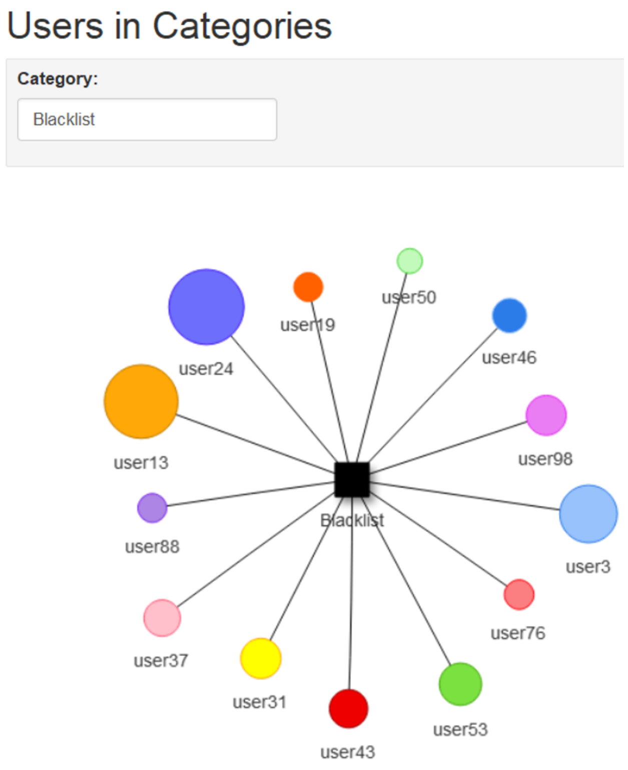

**Fig. 10.** Liable

We also incorporated two workforce planning visualizations. The first "Daylight," quickly shows the analyst when his users are active. Fig. 11 shows the progression of each of the three views. By leveraging alpha components of colors, the first view on the left shows you your entire employee population's schedule in unison. More transparent areas indicate that less users are active at that time. The visualization moves to progress to the right block where you can see each employee's working time individually. The notion draws from a basic pie graph, where a full circle would equal twenty four hours. The left side of the x axis indicates 6am and slices ending on the right side of the x axis indicate that they stopped working at 6pm. The visualization can be changed to either show the time as recorded on the machine, so the analyst would just see whether his employees are active in the day or night, or account for time zones if they wanted to investigate what users were active during the true time of a security incident.

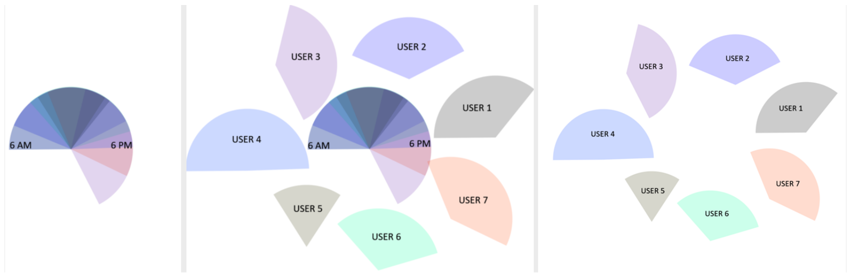

**Fig. 11.** Daylight

Our second workforce planning animation delves into the employee's location. The top map plot shows the clusters of employees clearly with a circle that augments based on how many employees are in each location. It would be able to clearly denote if a company has a more east coast or west coast culture or give a quick pulse of where employees are during a crisis.

The migration view of "Diaspora," focuses on just the subset of employees that move during a period of time. If there were a lot of connections, it would be insightful for revising travel policies. Also, if an employee crosses a border, compliance an HR professionals would have a method of following up on whether proper immigration, export compliance, or contractual obligations were satisfied during the employee's movement.

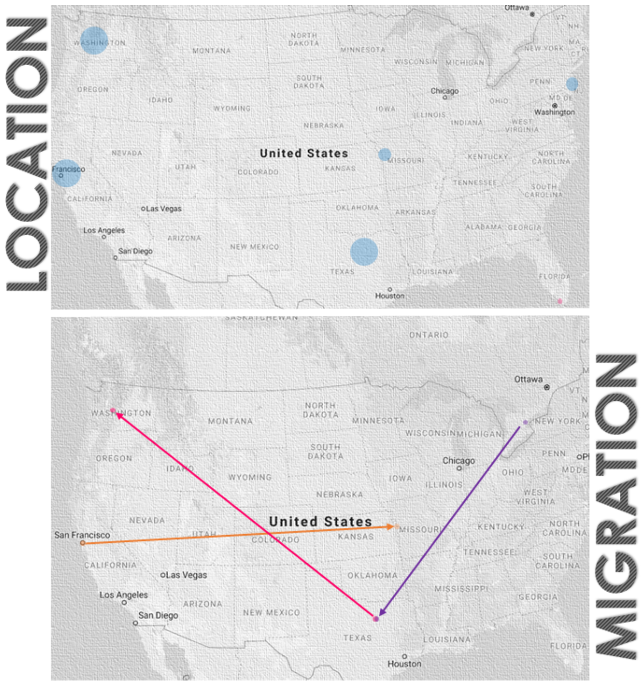

**Fig. 12.** Diaspora

## 12 Visualization Theory

When deciding how to visualize our results, we borrowed a lot from marketing theory. There is a reason why marketing materials are constantly changing their approach, syntax, color scheme, format, etc. We technically could present every visualization in the form of a line graph, but redundantly formatted information is often glazed over and the nuances that the visualization is trying to convey remains unnoticed. For this *Insight* utilization case, trying to convey the health of a network, we had to make sure that each topic/assertion was presented in a different way so the end user would mentally categorize them as unique, but individually important pieces of information. This was particularly challenging for our topic because there were many vectors that our data gathering process records that we did not want to oversimplify by not including them. However, an over ambitious visualization, as can see below, captures a majority of the vectors, but ends up confounding the problem and voices no message. Therefore, we had to be strategic in splitting our analyses into multiple visualizations with clear coherent thoughts/use cases.

We also had to take into consideration who our audience was. The primary user would be the security analyst at an enterprise, but we wanted them to be able to utilize the same tool to both investigate their environment as well as demonstrate it. To satisfy this need, each visualization was created to be

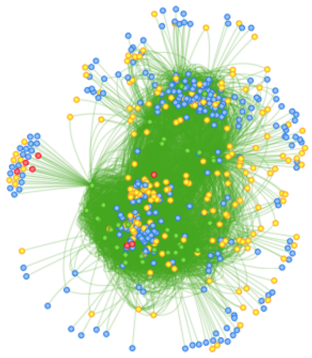

**Fig. 13.** Failed Visualization

scalable and demonstrate both high and low level information in an intuitive manner; generally, by scrolling inward or outward. This allowed for there to be an uninterrupted stream of communication across the hierarchy of a company. This method also reduced the number of visualizations needed to pull together to convey a thought, because each was overloaded with multiple comparisons. We also wanted to make the visualizations entertaining and unique from each other, so that the cognitive classification of the information would be pre-designated for the consumer.

Lastly, we took into account the common failures of dashboard design, primarily those that relay on a hyper dependence of software [12]. There is a common misconception that advanced software will also handle the design and communication for you. Designers can fall into their own marketing traps that focus on flashy software that may dazzle or entertain a viewer, but provide no inherent value. We made sure to split our visualizations across multiple tools and languages, some of them very primitive, so we were only emoting and communicating precisely what we intended and nothing more.

## 13 Analysis

The resulting *Insight* solution addresses cybersecurity concerns for enterprises from an endpoint level. The number of endpoints in an enterprise can vary greatly, but since each implementation will only be directed at a singular enterprise, there is a lot of homogeneity in the overall data to be capitalized from. By using endpoint data, we demystify cybersecurity analysis by interpreting basic values that most users already understand. The accessibility of *Insight* is the most crucial component because it allows executives, who can pivot and reallocate resources for a concern, to be informed enough to make those decisions. Although the input vectors into *Insight* are not intricate, the ability to understand what is normal and determine and diagnose an outlier is undeniably an effective method of predicting a change in performance. A security incident can be most succinctly characterized as just that, a change. There will never be a singular correct answer to the exact correct CPU utilization across all enterprises, the number of programs they should be allowing, or even the types of programs that are acceptable, but enabling enterprises to unveil what their optimal definition of normal is, provides both reactive and predictive solutions for enterprise cybersecurity.

## 14 Conclusion

Data science is considered to be a hybrid of four major topics: Statistics, Programming, Domain Knowledge, and Visualization. While the utility of the three are intuitive, because you cannot gather, ingest and analyze big data without them, the last is where the true value of the *Insight* software is found. One of our professors is loosely quoted to state: "If you can't communicate an idea, you don't have one." Visualization is essentially the too long didn't read (TL;DR) of big data. Considering the cost of a breach average more than $140 per record, and the number of records held by a company infinitely increasing. Enterprises no longer have the luxury of not knowing what is happening on in their systems. *Insight* drastically reduces the administrative burden gauging the health of enterprise end points by visually communicating Indicators of Compromise in an unambiguous manner.

The resulting visualizations were designed to be easily interpretable at higher levels of the organization yet contain enough information to satisfy those with more domain knowledge. This framework is adaptable and can be adjusted to visualize the data for high level anomaly detection or it can be scaled down to a lower level that helps identify minor discrepancies for subject matter experts. As a result the visualizations are regressive, ensuring that we didn't have a large breadth of visualizations that could be encumbering to a user, but rather a curated list of views that only exist for the most critical use cases. Lastly, the visualizations as a package offer additive insights, when combined in unison and allow the analyst or executive direction in exploring their overall IT infrastructure. By adopting this mantra, we can conclude that visualization is an incredible tool for reporting and communicating large amounts of information to a diverse crowd. Looking forward, we can use the information from the analysis to recommend optimizations to the *Insight* application such recategorizations of IOC programs as either whitelist or blacklist depending on how their metadata correlates with programs in either categories. Additionally, a single, interactive dashboard can be implemented after testing and receiving adequate feedback.

## References

1. Ji, S., Ma, H., Liang, Y., Leung, H., & Zhang, C. (2017). A whitelist and blacklist-based co-evolutionary strategy for defensing against multifarious trust attacks., Applied Intelligence, 47(4), 11151131. doi:10.1007/s10489-017-0944-x
2. Gregory-Brown, B., Harp, D. (2016). Security in a Converging IT/OT World., A SANS Whitepaper, 9-21.
3. Scneier, B.,Ranum, M. (2011). Schneier-Ranum Face-Off on whitelisting and blacklisting-Information Security Magazine., www.techtarget.com
4. Service Now(R), (2018). Kingston Security Incident Management., Service Now Documentation, 534-659
5. Bacik, S. (2011). Whitelisting., Information Systems Security, www.infosectoday.com
6. Ponemon Institute, (2017). Cost of Data Breach Study: Global Overview., Ponemon Institute LLC
7. Kent, K. and M. Souppaya, (2006). Guide to computer security log management., Tech. Rep. 800-92, NIST Special Publication
8. Swift, David (2010). Successful SIEM and Log Management Strategies for Audit and Compliance., SANS Institute InfoSec Reading Room
9. Turcotte, M. (2017). A. Kent and C. Hash, Unified Host and Network Data Set., in ArXiv e-prints
10. Mertens, Xavier (2017)., Whitelists: The Holy Grail of Attackers,, SANS Internet Storm Center., Sans ISC
11. Economist Intelligence Unit, (2017). The Meaning of Security in the 21st Century. The Economist, The Economist Newspaper, Palo AltoNetworks
12. FEW, S. (2016) Information Dashboard Design: The Effective Visual Communication of Data. O'Reilly Media, Inc.
13. EY Center for Board Matters, (2018). Cybersecurity Disclosure Benchmarking., EY.
14. Economist Leaders, (2017). Regulating the Internet Giants: The Worlds Most Valuable Resource is No Longer Oil, But Data. The Economist, The Economist Newspaper
15. Sidgwick, Henry (1907). The Method of Ethics. Reprint. NY: Dover (1981). Archived from the original on December 9, 2007
16. Johnson, M. (2005-11-03). Aristotle on Teleology., Oxford University Press
17. United States, Congress, House. United States Code. Title 18, section 2702, Office of the Law Revision Counsel, 14 Jan. 2017