SMU Data Science Review

Volume 2 | Number 1 Article 16

2019

Analysis of Computer Audit Data to Create Indicators of Compromise for Intrusion Detection

Steven Millett
Southern Methodist University, smillett@smu.edu

Michael Toolin
Southern Methodist University, mtoolin@smu.edu

Justin Bates
Colorado Technical University, Justin.Bates@ngc.com

Recommended Citation

Millett, Steven; Toolin, Michael; and Bates, Justin (2019) "Analysis of Computer Audit Data to Create Indicators of Compromise for Intrusion Detection," *SMU Data Science Review*: Vol. 2: No. 1, Article 16. Available at: https://scholar.smu.edu/datasciencereview/vol2/iss1/16

Analysis of Computer Audit Data to Create Indicators of Compromise for Intrusion Detection

Steven Millett[1], Michael Toolin[1], Advisor: Justin D. Bates PhD[2]

[1] Master of Science in Data Science, Southern Methodist University,
6425 Boaz Lane Dallas, TX 75275 USA
{smillett, mtoolin}@smu.edu
[2]Northrop Grumman
1 Space Park Dr.
Redondo Beach, CA 90278
Justin.Bates@ngc.com

Abstract. Network security systems are designed to identify and, if possible, prevent unauthorized access to computer and network resources. Today most network security systems consist of hardware and software components that work in conjunction with one another to present a layered line of defense against unauthorized intrusions. Software provides user interactive layers such as password authentication, and system level layers for monitoring network activity. This paper examines an application monitoring network traffic that attempts to identify Indicators of Compromise (IOC) by extracting patterns in the network traffic which likely corresponds to unauthorized access. Typical network log data and construct indicators are analyzed to predict network intrusion. Based on these indicators, a fitted model was created demonstrating which indicators best predict an intrusion event. In the end we found that XGBoost provided the best accuracy and f-score for our model fit. The IOCs that best predicted an intrusion event were associated with newly recorded events, network traffic, and DNS events.

1 Introduction

Cybersecurity is a major focus for companies due to an increasing number of attacks targeting their data and computer systems. Cybersecurity is the field that is focused on securing information systems and the data that resides on them. Data is quickly becoming the most vital resource to companies, making it an attractive target for cybercriminals. Many recent high-profile attacks attempt to exfiltrate sensitive information, as in the case with the OPM data breach in 2015 or attempt to make data unavailable as in the case of the WannaCry ransomware attack in May 2017. Although many tools are available to cybersecurity and IT professionals to identify or prevent these attacks, implementation and maintenance of these tools require companies to invest in large teams to manage the applications or require specialized expertise that may be cost-prohibitive for small companies. Our goal was to use the standard logs typically generated by IT environments to reduce the amount of time needed in identifying unauthorized access and minimize any damage done by an attack.

Our approach to finding indicators of compromise (IOC) is by either abstraction of audit data or applying classification techniques to the aggregated audit data to create models that detect positive attack patterns. We created easily understood indicators for cybersecurity analysts to investigate possible attacks or provide enough insight for investigators to know where to further investigate any possible attacks. We can make security professionals more effective by providing tools that add insight using the systems that are already at their disposal while reducing the effort needed to gain this understanding. Effective means of intrusion detection involves abstracting known malicious material into indicators of compromise. Indicators of compromise are cyber events that strongly indicate a possible attack on a computer or a network. IOCs alert security administrators or auditors to investigate security events, but if the rules configured for identifying IOCs are set incorrectly then intrusions can be missed, or the security team can receive false positives alerts. Several factors are considered when gauging if an IOC is a true positive. These factors include multiple IOCs occurring in a short period, the IOC is associated with high-value equipment, or an event registered as a critical IOC.

The idea of an IOC would make it seem there is a common signature for computer network intrusions, but as seen by the significant delays in detecting network intrusions, this assumption is wrong. From research done by IBM, the mean time to identify breaches was 197 days, with 69 days being the mean time to contain.[1] This delay in detection could be attributed to many factors including lack of evidence or poor staffing. A breach can be attributed to improper system configuration or an attack by a cybercriminal. Breaches due to improper configuration are all too common, sometimes with significant repercussions as in the case of the breach of 123 million records from Alteryx in 2017,[2] unclassified intelligence data found in 2017,[3] or top-secret data from the United States Department of Defense found in 2017.[4] The attacks associated with initial misconfigurations are beyond the scope of this paper due to their lack of persistence. While some breaches can be attributed due to misconfiguration, more damaging attacks occur due to the efforts from an attacker that is actively exploiting temporary access with the goal of maintaining that access and moving laterally through the network.

Although no two network intrusions are the same, many follow a similar pattern. As outlined in the National Institute of Standards and Technology (NIST) publication 800-

[1] "2018 Cost of a Data Breach Study: Global Overview," Ponemon Institute LLC, 2018. [Online.] https://databreachcalculator.mybluemix.net/assets/2018_Global_Cost_of_a_Data_Breach_Report.pdf [Accessed 19 October 2018]

[2] "Alteryx data breach exposed 123 million American households' information," Los Angeles Times, 22 December 2017. [Online]. Available: http://www.latimes.com/business/technology/la-fi-tn-alteryx-data-breach-20171222-story.html [Accessed 19 October 2018]

[3] "Defense contractor stored intelligence data in Amazon cloud unprotected," ArsTechnica, 31 May 2017. [Online]. Available: https://arstechnica.com/information-technology/2017/05/defense-contractor-stored-intelligence-data-in-amazon-cloud-unprotected/. [Accessed 19 October 2018]

[4] Dan O'Sullivan, "Black Box, Red Disk: How Top Secret NSA and Army Data Leaked Online," UpGuard, 28 November 2017. [Online]. Available: https://www.upguard.com/breaches/cloud-leak-inscom. [Accessed 19 October 2018]

115 which details the penetration testing methodology employed by government contractors, there are four primary stages to a penetration test: Planning, Discovery, Attack, and Reporting. The planning and reporting phases are beyond the scope of this paper as they mainly deal with the rules of engagement and follow-up to the penetration test. The discovery phase, also known as reconnaissance, and the attack phase involve the technical aspects of network intrusions where an attacker actively runs probes across the network to gain further access. NIST 800-115 lists four steps of the attack phase: gaining access, escalating privileges, system browsing, and installation of additional tools [1].

Currently, intrusion detection as a field is more art than science. Intrusion detection involves investigative techniques that are used to analyze possible malicious activity after an intrusion has already taken place. Depending on the circumstance, intrusion detection involves either responding to detected IOCs or, worse yet, investigation IOCs after initial evidence of an intrusion is found. For example, in federal information systems, requirements are outlined by the National Institute of Standards and Technology (NIST) in multiple documents detailing the requirements for intrusion detection and prevention systems. Due to the many different ways that an attacker can gain access to a system, different system logs are monitored for any signs of abnormal activity. An intrusion detection system is any software that, "monitor[s] the events occurring in a computer system or network, analyze[s] them for signs of possible incidents," while a prevention system takes additional steps to stop any incidents while they are occurring [2].

Collecting the information necessary for effective intrusion detection means data needs to be collected from multiple areas in the network from a variety of devices. The amount of logging data that is generated in a standard corporate information system can be staggering and working with this data can be difficult. Previous work in creating frameworks for intrusion detection focused on using datasets that had already constructed features from the original system data. This abstraction can be unrealistic since this feature creation is not done automatically by systems that generate these logs, rather done before the dataset was released to the public. We created our model to construct features from data that is nearly identical to what is found in system log files and discover and identify significant indicators of an intrusion.

Solving the problem of identifying significant IOCs is done by following a three-step methodology to ingest the data, create relevant features (IOCs), fit our model, and interpret the features on the fit of the final model. The dataset is made up of log data, which is not easily interpreted by a classification model. The first step is to split the dataset into train, validation, and test datasets thereby ensuring our trained model is generalizable beyond what is captured in our sample. Next, we need to create features from the data by identifying anomalous sequences in the logs; this requires that we extract specific patterns for a given source host, such as repeated failed logins, new DNS requests, or the number of bytes sent, all of which depend on the source of the log data. After feature construction, the data is preprocessed which includes normalization of the numbers and removing any null values.

Step two of the methodology is modeling which is applied once all features have been constructed, nulls removed, and values standardized. Modeling is done by using a technique known as k-fold cross-validation, which is a method of improving model accuracy by dividing the training data into k-number of separate folds which are used to reduce bias in the final trained model. Fitting of the input data to the intrusion events was done with four different classification models, including random forest classification, logistic regression, SVM, and XGBoost. Logistic regression, random forest classification, and SVM are well-known classification models, while XGBoost is a newer algorithm that has gained popularity due to its high accuracy and relative ease of implementation. Logistic Regression is like linear regression in that each feature contributes multiplier β to the dependent variable, the difference being that the dependent variable is transformed using a logit function, so the final value is bound between 0 and 1. Random forest classification is a technique that builds off decision trees where branching choices are made on different values of the features. Support vector machines (SVM) is a method of dividing the data such that a hyperplane drawn in an n-dimensional space (where n is the number of features) such that the hyperplane best separates observations with different labels. XGBoost is a variation on boosted decision trees that has gained much popularity due to its speed and accuracy. Accuracy was used as the primary metric to compare the predictive ability of the different models.

The final step of the methodology is applying our classification model to the test set using the final training models and interpreting the features' influence on the strength of the model. This step involves interpretation and analysis using background expertise to provide context to the significant IOCs. With interpreting the IOCs, we also measure the ability of our model to predict future attacks effectively. It is this ability to predict future results that demonstrates the overall effectiveness of our model.

Through the rest of the paper, we discuss the following items. In Section 2 we review similar research material related to the topic in this paper. In Section 3 we review the general breakdown of intrusions and understand how these could present different signatures. Section 4 breaks down the different audit logs that make up the dataset and how this data is interpreted. Sections 5 through 7 break down the different steps of our methodology and how it is applied to our dataset. Section 5 we construct our features and preprocess them to ensure we are meeting the assumptions of our classification models. In section 6 we model the results and compare our chosen modeling techniques to our testing metric. Section 7 analyzes the IOCs and examines which play a more significant role in predicting intrusions. Section 8 covers the ethical and societal impacts of data collection and security breaches. In sections 9 and 10 provides the conclusions of our research and proposes future research topics.

2 Literary Review

We looked at previous research done in the field of intrusion detection and feature extraction. Much of the previous research was focused on the Knowledge Discover and Datamining (KDD) Cup 1999 dataset which is a derivative of a dataset from the 1998 DARPA Intrusion Detection Evaluation Program [3]. There has been extensive research into this dataset since a significant portion of the feature construction was already performed on the original dataset from the DARPA program. Much of the preprocessing was previously done, and only analysis into the existing features was necessary. There has been analysis into using network theory to conduct feature extraction in comparison to principal component analysis (PCA), where it was found there was a 1% detection rate compared to PCA, yet network theory was 13% more efficient [4]. Work has been done on the KDD Cup 1999 dataset using different classification techniques to conduct feature selection with multiple machine learning techniques including support vector machines (SVM), Classification and Regression Trees (CART) and BayesNet [3]. Additional papers implemented a combination of dimensionality reduction and classification to determine the best fit of machine learning techniques, including SVM with PCA [5], SVM with rough set kernel principal component analysis (RS-KPCA) [6], and neural nets with KPCA [7]. Both PCA and RS-KPCA are methods of dimensionality reduction that transform the input values using linear algebra to extract the significance of each variable has in explain the total variance across all variables. The fractional share that each variable has in explaining total variance is then used to create new variables, which is then used to train a model. While dimensionality reduction can create powerful models, we did not use any such techniques in this paper because these data transformation significantly diminish interpretability of the final model and make it difficult to understand which features have greater importance in the final model.

Work has been done on creating frameworks to identify anomalous data in sequence and categorical information. Previous research has been done in identifying episodes in sequences and working to predict future behavior by Mannila, Toivonen, and Verkamo [8]. Jain, Duin, and Mao work on codifying multiple pattern recognition techniques and demonstrate a model for train-test split that can assist in identifying feature importance [9]. This research for anomaly detection was built upon by Lee, Stolfo, and Mok when they created their data mining framework to identify significant features in the KDD Cup 99 dataset [10]. We want to build on this work of framework construction and pattern recognition by constructing our own features from actual log data.

3 Cyberattack Methodology

When a computer network comes under cyber-attack, this typically is not a one-time event. A malefactor takes multiple steps to probe a network and discover its vulnerabilities. Different models exist describing the methodology used in a cyber-attack. One model described in NIST 800-115 is a four-stage penetration methodology. As the name suggests, it has four stages: Planning, Discovery, Attack, and Reporting. The components of each stage are similar to other models. One such model with similar components used to describe the different stages of a cyber-attack was developed by Lockheed Martin, the Cyber Kill Chain [11]. The cyber kill chain is used by cybersecurity teams to understand the methodology of attack from advanced persistent threats. The framework is used to compare the different techniques that a sophisticated attacker would use in trying to get to their objective. For our purposes, it provides a good set of steps that an attacker would follow when trying to exploit system access. We can use this model to describe the steps a malefactor would take to breach a network:

1) **Reconnaissance –** This is the stage where different types of vulnerabilities are identified. A company's web site is examined by scanning network ports and identifying services operating on various ports. Employee email addresses are obtained,

or other methods of identifying employees are used through professional relationships.

2) **Weaponization -** This is the stage where the virus, worm or other malware is embedded into a payload such as a Microsoft Office document or a PDF file. The weapon has not been deployed at this point; it is simply prepared.

3) **Delivery** - This is the stage where the malefactor delivers the weapon to the intended target. Any data transport mechanism could be used for this, such as email attachments, website downloads or even physical media such a USB thumb drive.

4) **Exploitation** – In this stage some vulnerability is exploited, providing access. It could be as simple as a user clicking on a link in an email or running some code on a USB drive. "Zero day" is a term used to refer to this stage, as it is the first time some code is run to exploit the system.

5) **Installation** – The malware or trojan is installed on a system giving the malefactor a backdoor entrance to the system. This allows the adversary more or less continuous access to the system.

6) **Command and Control (C2)** – The installed malware creates a connection outside the system providing some level of control of the system. This connection can be an external internet server or some other host system. At this point, the intruder can issue commands to the malware for execution.

7) **Actions on Objectives** – Once command and control is established, the adversary can now accomplish their goal. Whether this is stealing data, deleting or modifying information, denying access to critical data or just using this resource to gain access to other network resources, the intruder has complete control of what they wanted to do.

This is a detailed set of steps that are involved in gaining system access, and not all of this would be evident from our system logs. We can see from this model that there are many different aspects of the information system that are involved when an attacker is trying to gain unauthorized access and accomplish their goal.

4 Dataset and Modeling Experiments

Intrusion detection as a field is not new; tools have been created for decades to detect possible malicious activity. Software packages like Snort and Bro were traditionally installed at specific segments of the network to capture relevant network information that would indicate a possible compromise. Both Snort and Bro are open-source network intrusion tools that implemented static detection methods through defined rule sets.[5] Over time these tools have become more sophisticated with improved rule sets, but many still rely on signatures to issue indicators of compromise. These signatures detect if certain phrases are in the body of the network packet, if the packet was a certain size, or if there is a certain number of patterns in the network traffic, such as too many failed logins. The effectiveness of signature-based tools is limited to how complete the database of signatures is, but even with an up-to-date library, a zero-day attack can bypass the most sophisticated signature-based system. This has led to significant investment in adaptive intrusion detection with an emphasis on the use of machine

[5] "Snort, Suricata and Bro: 3 Open Source Technologies for Securing Modern Networks," Bricata, 2018. [Online.] https://bricata.com/blog/snort-suricata-bro-ids/ [Accessed 20 January 2019]

learning. In this section, we analyze the chosen dataset and discuss how we extract features from each log type to detect an intrusion.

4.1 Dataset Description

We are using the Los Alamos Multi-Source Cyber-Security Events dataset. This data set is made up of different types of computer audit logs that were recorded over the course of 58 days, from multiple sources that would be found in a traditional computing network environment. During these 58 days, an approved penetration test was conducted against the network with all malicious activity identified. This approved penetration test was conducted by a cybersecurity red team, which is a group that is separate from the normal cybersecurity group whose role is to test the security of the computer security infrastructure. Often a red team would be used to conduct tests against all facets of an organization's security, but for our purposes, we are only looking at the identified computer-based events as previously identified in the dataset. The events recorded during the test are used to train a model whose goal is to help future implementations of Intrusion Detection Systems (IDS) correctly identify and alert cybersecurity teams of possible unauthorized access to this network. The data set that is being used is approximately 100 gigabytes, but only a small fraction of that are the red team events. Dealing with this imbalance between the predicted outcome and the size of the complete dataset would mean that additional considerations would need to be taken during modeling. In small to medium environments, terabytes of auditable data can be generated in a single day which would need to correlate across multiple systems with different indicators of a possible intrusion so while this dataset is large, it does not compare to what many corporations have to deal with [12].

The data from the logs can be broken down into two parts, events that occur on the computers (host-based events) and events on the network (network-based events). Events that occur on the computers would be dependent on the type of environment, i.e., Windows or Linux, while network-based events should be mostly environment independent. In the next two parts, we cover the logs from these two groups and what information these logs are capturing. We also look at how abnormal activity would appear in each log type. The dataset was collected in four separate logs: Authentication logs, DNS logs, Network flow logs, and Red Team logs. The logs and the information tracked in them are discussed below for the major sections. The red team tracking logs are unique since these are used as the labels for our supervised learning. The red team logs are closest in format to the authentication logs since they track the authentication events associated with red team activity, but they do not track the same fields that are found in the authentication logs.

Table 1 - Contents of Red Team Logs

Feature	Description
Time	Time in seconds from the beginning of the recording period
User & Domain	Anonymized user and domain information of a known red team authentication request.
Source Computer	The anonymized name of the computer that is the source of the authentication request
Destination Computer	The anonymized name of the computer that is the destination of the authentication request

4.2 Windows Event Viewer Analysis

The built-in method of recording audit events in a Windows environment is the Windows event logs which track application, system, and security event information for later auditing. Depending on the type of log, different information is recorded making manual auditing of these events difficult due to formatting inconsistencies. Depending on the source of the auditing information this data is recorded in different locations. In the case of the authentication log data, this would have been recorded on the Domain Controller (DC) or on the local computer depending on whether the user was authenticating to the DC.[6] The foundation of Window's domains is built upon the Active Directory Service which is a technology built on Kerberos, Lightweight Directory Access Protocol, and DNS. Kerberos provides authentication and authorization while Lightweight Directory Access Protocol (LDAP) documents and categorizes the objects making up a domain.

Understanding how Kerberos authenticates hosts on the network is critical to extracting features from the authentication logs. Table 2 breaks down the information contained in the authentication logs.

From the authentication data, we created features based on different statuses of the logon event, such as failed logon. Also, we wanted to see how many people were logged on to a computer at a time. We expect these features to vary by time of day, so we processed the data to establish a baseline of use for a given time and day.

[6] "Monitoring Active Directory for Signs of Compromise", Microsoft Documents [Online.] https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/monitoring-active-directory-for-signs-of-compromise [Accessed 4 November 2018]

Table 2 - Contents of Authentication Logs

Feature	Description
Time	Time in seconds from the beginning of the recording period
Source User & Domain	The anonymized user and domain information that is the source of the authentication request.
Destination User & Domain	The anonymized user and domain information that is the destination for the authentication request
Source Computer	The anonymized name of the computer that is the source of the authentication request
Destination Computer	The anonymized name of the computer that is the destination of the authentication request
Authentication Type	Type of authentication
Logon Type	Type of Logon
Authentication orientation	Whether this is a Logon or Logoff
Success/Failure	Authentication was successful or failed

In this type of log, we are interested in the abnormal event types including repeated failed logins, users logging into computers that they do not regularly access and detecting if the same account is logged on multiple times at once. While these event types may not be a single indicator of a breach, with traditional account auditing, such events would often warrant further investigation.

Process execution and termination are recorded in event logs under the system or application logs. During this analysis, we conduct some sequence analysis and investigate whether a particular program was executed shortly before a red team event. Additional items of interest would be whether many events were launched or if a computer launched an application that it usually does not execute. The process log tracking this information is shown in Table 3 below. Process log data was interpreted using link analysis as explored by Lee, Stolfo and Mok [10]. We want to extract how many processes are running on a given source computer and how many new processes are launched on a source computer.

Table 3 - Contents of Process Logs

Feature	Description
Time	Time in seconds from the beginning of the recording period
Source User & Domain	The anonymized user name and domain of the source where the process exists
Computer	The anonymized computer name where the process exists
Process Name	The name of the process
Start/End	Defines if the process was starting or ending

4.3 Network and DNS Configuration

Networking infrastructure is the backbone of any corporate environment as few companies operate computers in isolation. Maintaining consistent network connectivity is essential to an information system, and the networking group often is the first group to be notified if there is an issue with systems communication. Outside of detecting intrusions, network traffic diagnostics are used for many issues like tracing traffic routing issues, high utilization of resources on computers, an improper configuration in network-based applications, and improper use of system resources. We looked at anomalous levels of activity from host to host using network traffic flow data. Table 4 contains the listing of the fields in the network traffic flow audit logs. The network traffic data that is continuous data were added together to get the total amount sent during the 5-minute sample of time.

Table 4 - Contents of Network Traffic Flow Logs

Feature	Description
Time	Time in seconds from the beginning of the recording period
Duration	Time in seconds the transaction took to occur
Source Computer	The anonymized name of the source of the network traffic
Source Port	The port number the source of the network traffic transmitted from
Destination Computer	The anonymized name of the destination of the network traffic
Destination Port	The anonymized port number the network traffic was destined
Protocol	The anonymized protocol used for data delivery
Packet Count	The number of packets transmitted in this transaction
Byte Count	The number of bytes transmitted in this transaction

DNS serves multiple purposes in a Windows networked environment. DNS stands for domain name system, and it provides an IP address when it is provided a hostname or URL. For example, when you type www.smu.edu into a web browser, the address 129.119.70.166 is returned by your local DNS server. When you need to access a file or mail server on the network in a Windows environment, DNS returns the proper IP address of the requested resource. It is this ability to return IP addresses or logical addresses based on hostnames that play a fundamental role in a Windows network. Since Windows tracks devices by name, it needs a way to resolve how to route the traffic over the network.

We want to conduct link analysis for DNS data as noted by Lee, Stolfo and Mok [10]. Link analysis tracks systems that are often working together, and we want to find patterns outside of those common links. For example, a user in the sales department may regularly access the file server where her tracking information is stored, but we want a red flag raised if she starts trying to access an application server that she has never used before. Table 5 lists the fields in the DNS log data file and what each field means. We want to look at how many DNS requests a given computer is making and how many new DNS requests the source computer made as features we are extracting from DNS.

Table 5 - Contents of DNS logs

Feature	Description
Time	Time in seconds from the beginning of the recording period
Source Computer	The anonymized name of the computer that requested a DNS lookup
Destination Computer	The anonymized name of the resolved computer of the DNS lookup

5 Step 1 - Feature Construction and Preprocessing

The first step in working with this dataset is creating features from the dataset. Due to the event-based nature of the data, each log has a different number of events for the same time. We extract the relevant metric that we want to measure against our label that we are interested in, which are verified red team events from the red team log data. The data is grouped for a given timeframe (in this case we are grouping initially by 5-minute intervals) and source computer, and then we extract the relevant metric for the given log. We are using a couple of different techniques depending on the type of log data as first described by Lee, Stolfo, and Mok [13]. Depending on the type of data we created features using either classification, link analysis, or sequence analysis. Classification is used to identify specific patterns in the logs as either binary or multi-label results.

For both the DNS and the process data we are using link analysis to measure two data points for each source computer; the number of new links not previously established for a given source computer and the total number of links. While it is easy to think of DNS traffic as being unrelated to process information, they share similar characteristics when trying to model abnormal behavior. For example, with DNS traffic we want to know how many computers on the network the source computer is trying to access, and we want to see how many new computers the source computer is trying to reach. This same thinking also applies to computers running on a source computer. We wanted to see how many processes are running on a given computer along with how many new processes are running. Abnormal numbers could be an indication of a virus or other code meant to exploit the system running on the source computer.

In the authorization dataset, we calculate the number of times that failed logons occurred and the number of logged on users for a given system. An increase in the number of failed logons could be an indication that a malicious user is trying to gain access to an account without knowing the correct credentials. An increase in the number of logged on users could be an indicator that the same account is logged on in multiple locations or multiple accounts are being accessed, which usually aren't at a given time.

Processing of the network flow data was the easiest as the original dataset was already in a form that represented metrics for data sent over the network. To get the desired values for the specific field, the sum of a metric was obtained for a given source computer for a 5-minute interval. For example, the packet count input column was summed up for a given computer for each 5-minute interval resulting in a feature containing the total number of packets sent by that computer.

After all the initial features are extracted from the raw datasets, the mean values for each column are calculated for all source computers in each given 5-minute interval. This was done due to the significant computer resources that would be required to calculate a model for all 17,684 computers across the time frame. Rather than looking at a red-team event on a given source computer, we are looking for a red-team event on the network. Once a red-team event is identified further investigation can be done to determine which computer exhibits the characteristics of the uncovered red-team event.

Once this data is aggregated, a model is taken for all data to determine the usual pattern for a given feature. For example, when examining the number of logged on users, more people would be logged on at Monday 12:00 PM than would be on at Monday 12:00 AM or on Sunday at 12:00 PM. We use the Facebook library prophet to train a model on our dataset on the usual pattern of usage day-to-day and week-to-week. We subtract this regular usage pattern from each feature, so we are left with only the outliers for a given period and feature. This final set of features is used to train our classification models.

6 Step 2 - Modeling

After the data is formatted consistently and the hosts are identified, we can run preprocessing and modeling on the data. Some data has very large ranges such as the network traffic feature, while features such as failed logons have a much smaller range. We do not want one feature having an over-weighted influence on our results, so all features are normalized before running through our classifier.

Initially, we separated our data into three segments; train, validation, and test sections which can be seen in Figure 1. Our model is initially trained on the train data to fit our classification algorithm and parameters. We then want to select our model from testing our model against our validation dataset. The reason this data is separate from our training data is that we want to generalize our model and avoid overfitting to our training dataset, while at the same time we want to avoid training to our test set. The test set should be the final test to verify that our model is fitted to the assumptions of our classification model and parameters.

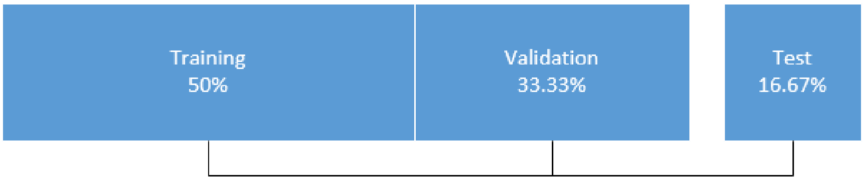

Figure 1 - Train, validation, test split of our dataset into different portion for proper validation and model selection.

As part of the classification process, we used k-fold cross-validation to ensure the trained model is generalizable beyond the scope of this dataset. K-fold cross-validation is a technique of splitting our data into equal segments called folds and using 1-fold as a testing set while the remaining folds are used to train the model. This is done k-times for the input data and is done to reduce overfitting of the model on the dataset. We use a modified version of k-fold cross-validation known as stratified k-means. This is used because of the high frequency of observations where there is not an occurrence of red team activity. We need to oversample the times that red team activity is present in our predictions to ensure we have some samples of these occurrences in each fold.

We experiment with several classification methods including logistic regression, Random Forest Classification, SVM, and XGBoost. We were looking for reliable classification methods that also gave an insightful interpretation of feature impact for the fitted model.

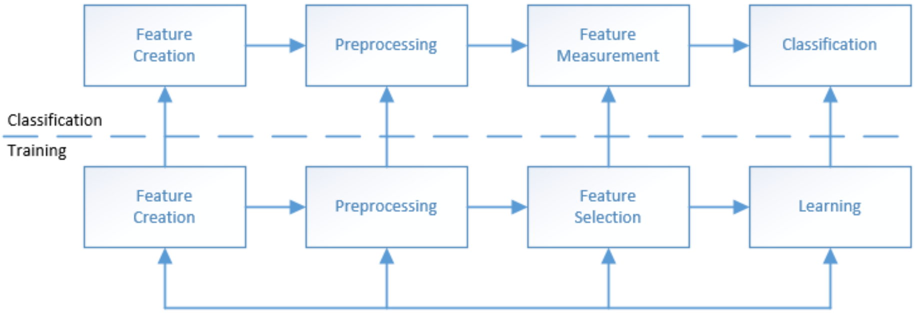

Figure 2 - Data flow process for feature creation.

Figure 2 shows a model outlining the dataflow process from feature creation to modeling. This model is based on the model first proposed by Jain, Duin, and Mao as a method of creating a system for statistical pattern recognition [9].

Referencing the process in Figure 2, we are concerned with the last two steps for each row, the feature selection and learning portions of feature creation. This feature selection and training was done using a pipeline in Python allowing for modular code segments to be changed out for systematic retesting of machine learning models. The pipeline allows for multiple classifiers to be tried at once which simplifies final analysis as everything is saved into an easy to explore variable. We use grid search to optimize the training of the pipeline and to fine-tune the relevant hyperparameters of the classifiers. Grid search allows testing of many variations of tuning parameters in the pipeline to find the best fit for our data.

Logistic Regression can be thought of as the most straightforward and easy to understand classifier from the group tested. Logistic regression builds off linear regression, where the predicted y value is limited to a value between 0 and 1 through the transformation using a sigmoid function.

The Support Vector Machine (SVM) is a recent development in statistical analysis. The support vector machine attempts to classify an observation into one of two categories. It does this by calculating a hyperplane which separates the classes of observations into two classes. Depending on which side of the hyperplane an observation is located determines which class it is assigned.

The techniques Random Forest Classifier and XGBoost build off of the decision tree model. Decision trees work to predict a label by following classification rules that create branches in the model. A classification rule is created by selecting a value or range of values of a feature or multiple features and dividing the data along that branch. The chosen branch can lead to a classification label (leaf) or to other branches which would eventually terminate in a final leaf. Decision trees are one of the most intuitive classification models for interpretation but can be prone to overfitting.

Random Forest Classifier is a modification of decision tree classifiers, which creates multiple copies of the decision trees and averages the results between the multiple copies to come up with a final model that is less biased and not as prone to overfitting as a single tree.

XGBoost builds off of decision trees by using a gradient boosting approach. This approach uses gradient descent to minimize the loss function as new models are added. This process continues adding new models until the loss function is minimized and no improvements in the model can be made.

Selecting the optimal model involved plotting the mean test score for each round of model fitting with the pipeline for a given model. We grouped all the test scores by classification model and plotted their respective box plot. The benefit of the plots is that they show the total range of the given model and the median of the accuracy of the model. We selected XGBoost classifier as the model to use to predict for our final classifier since it has the highest average accuracy and the smallest range for accuracy scores. We wanted a model that is robust to changes in the dataset and less likely to provide widely varying prediction results depending on changes to the input data.

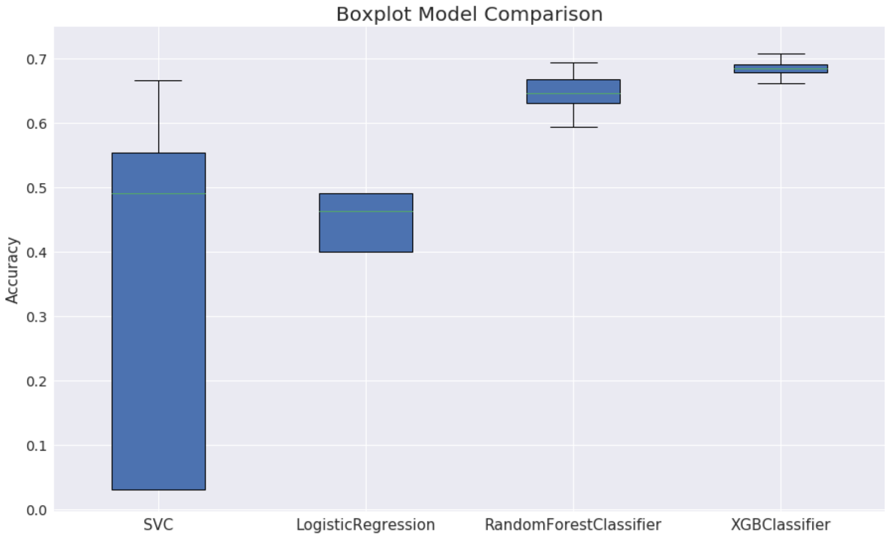

Figure 3 - Boxplot model comparison of the different machine learning algorithms tested.

In Figure 3 we showed the comparison of the accuracy of the different classifiers based on the training process. Due to the imbalance between the non-events to positive red team events we want to run further analysis to determine the effectiveness of identifying true positives while reducing the false negatives, known as the recall, and the ability to identify true positives while reducing false positives, known as precision. In Table 6 - Model Comparison we show the classification reports of each of the classifiers working on a separate validation dataset. We are testing against this new dataset to avoid selecting a model that is overfitted to our training data.

From the entries for SVM and Logistic Regression we see that even if zero true positives are predicted the model would have a 96.7% accuracy. Based on these results neither SVM of Logistic Regression are valid models since they have a zero for both precision and recall.

Table 6 - Model Comparison

Model Comparison Classification Report

Classification Model	Accuracy	Precision	Recall	F-Score
SVM	96.7%	0.0%	0.0%	0.0%
Logistic Regression	96.7%	0.0%	0.0%	0.0%
Random Forest Classifier	96.6%	55.0%	5.0%	10.0%
XGBoost	96.7%	46.0%	15.0%	23.0%

For a final comparison, we plotted the ROC curves for our four models using the validation dataset in Figure 4. The ROC curve gives a visual representation of the model classification report where it plots the relationship of each model's sensitivity (true positive rate) against its specificity (true negative rate). These plots help to reinforce the decision to use XGBoost as the optimal classifier for our final model.

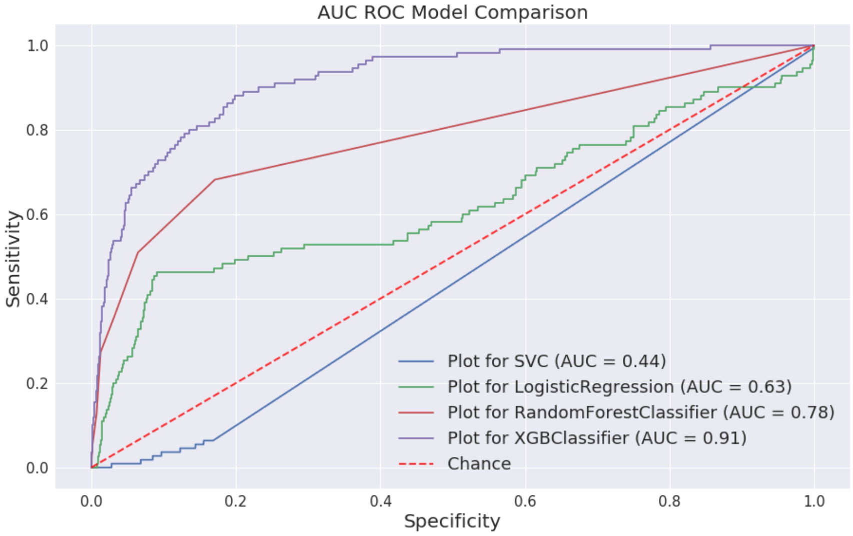

Figure 4 - ROC curve model comparison shows the performance of the different algorithms to reduce false positives and optimize true positives.

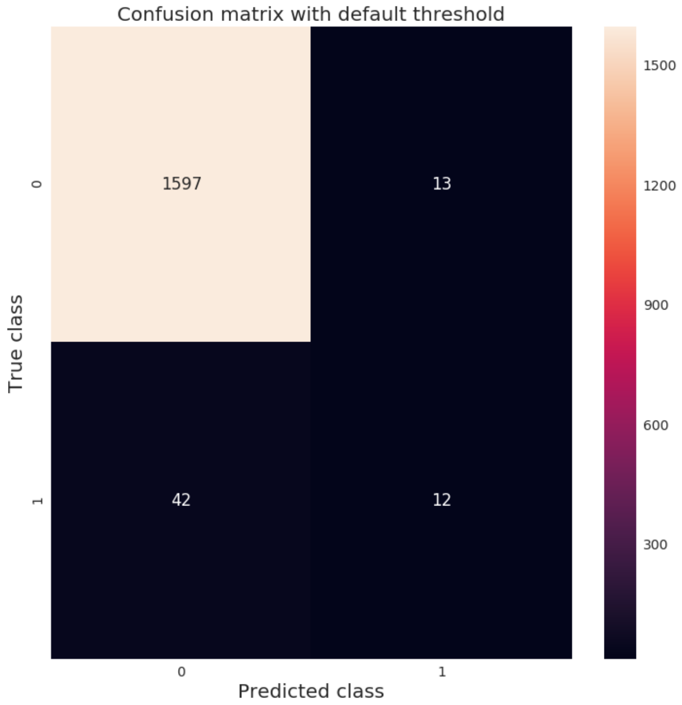

Figure 5 - Initial confusion matrix without optimization for our decision threshold.

In Figure 5 we present the confusion matrix for the XGBoost model. The confusion matrix is read as follows; the bottom right of the matrix shows how many times we predicted a red team event when one actually happened (12). The top left of the matrix shows how many times we predicted the lack of a red team event when no event happened (1597). Each of these events is showing how often we correctly predicted the true state and make up or accuracy score. The top right quadrant shows how many times there was not a red team event, but we predicted one had occurred (13). These are also known as Type I errors. The bottom left of the matrix shows how many times a red team event did occur, but we failed to predict it (42). This is also a Type II error. When predicting network intrusions, we want to be as accurate as possible, but we also need to limit the number of Type II errors. This situation means an intrusion occurred, but we failed to predict it. We need to raise the Recall score to do this. Ideally, we would also raise the Precision score which is measuring how often we predicted an intrusion when in fact none occurred. However, raising the Recall score is more important as we want to limit the number of intrusions we miss.

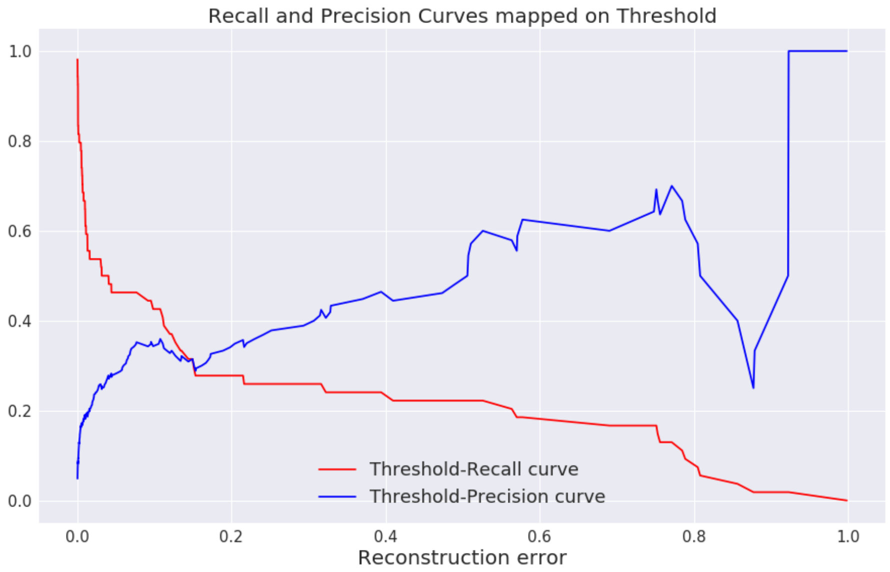

Figure 6 - Recall and precision curves show the optimal threshold where our recall and precision curves intersect.

Modifying the threshold hyper-parameter of the model results in the chart seen in Figure 6. this helps identify the best threshold for our model. We re-ran the model with the updated parameters and threshold set at 17%. The result is seen in the confusion matrix depicted in Figure 7. This updated model is doing a slightly better job at predicting when a red team event occurred (17), a true positive event. The other result is we have fewer false negative predictions (37), meaning we aren't incorrectly predicting a non-event when in fact a red team event has occurred. We have reduced our Type II error at the expense of increasing Type I errors.

Table 7 Recall Scores of Test Dataset

Recall score comparison of the test dataset

Threshold setting	Accuracy	Precision	Recall	F-Score
Initial setting @ 50%	98.5%	48.0%	22.2%	30.2%
Optimized setting @ 17%	95.5%	31.5%	31.5%	31.5%

Table 7 shows the increase in the Recall score. A side effect of this increase has resulted in a slightly lower overall Accuracy score and a lower Precision score. The changes in these scores are acceptable trade-offs to get the increase Recall score.

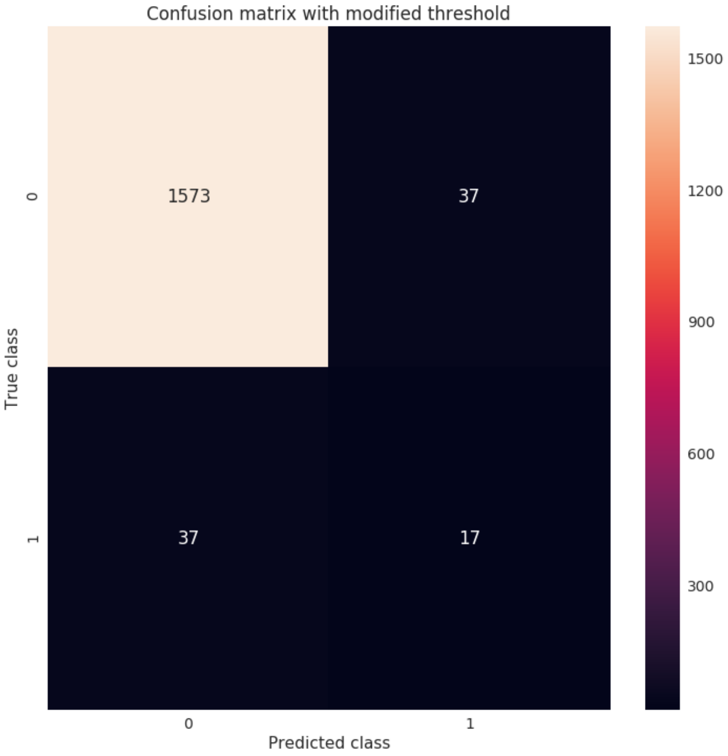

Figure 7 - Final confusion matrix showing the false positives and false negatives when our threshold is optimized.

7 Step 3 - Measuring Importance of IOCs

The final step for creating our model was comparing the relative weight of the features for each classification model and the relative score of each model. Now that we have a model that is fit to the data we extracted the individual importance of each feature to predicting likely red team events. In Figure 8 we see the relative importance that each feature has in the final model. The top two events are related to new processes and DNS requests made by compromised computers. This would be expected as a new activity that is outside of the normal operation for the user. The next two features, logged on and failed count, are associated with authentication events and the last of the top five significant IOCs are higher than usual DNS queries.

The top 5 events can be broken up into three major categories, greater than usual new process and traffic activity, increased user activity, and increased network traffic. The next four important features beyond the top five are all associated with increased network traffic. So additional user activity that is outside of the normal activity would be an indicator of compromise.

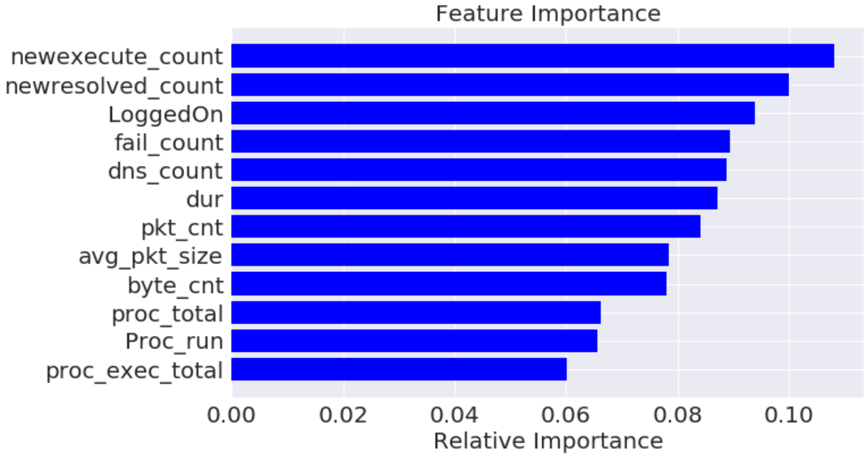

Figure 8 - Feature importance of the final model chosen from our machine learning model.

8 Ethical and Societal Impact

There can be significant impacts on individuals and companies should they fall victim to their data being compromised due to a security breach. As a result, governments are enacting legislation to ensure there are penalties for companies that do not take proper precautions in securing private data in addition to giving people greater control over their data. Penalties are centered around the lack of proper controls and mandating companies to reveal when a breach has occurred. To enforce new regulations companies are having to take great steps to capture and monitor user and employee data on their networks. This desire for individual privacy and required corporate security means that companies need to take care when implementing information security policies. We discuss the implications of these laws on big data, information security, and the impact on people.

8.1 Ethical and Privacy Implications of Corporate Data Collection

The right to privacy has been documented going back as far as an article in the *Harvard Law Review* by Samuel D. Warren and Louis D. Brandeis in 1890 where they stated "Recent inventions and business methods call attention to the next step which must be taken for the protection of the person, and for securing to the individual [...] the right 'to be let alone' [14]". Going as far back as the referenced Harvard Law Review article we have an expectation of privacy, which is now extended to all aspects of our lives including our online presence. However, this expectation of privacy is different when people are at home versus when they are at work. In the home and on personal computing devices, people may believe their data is not being snooped, but this viewpoint changes in the workplace. The majority of employees do not have a reasonable expectation of privacy in the workplace; rather, there is the expectation they would be given notice they are being monitored [15]. There is extensive case law siding with employers on the topic of employee privacy. Even though employees may understand their internet usage is being monitored in the workplace, and it is legal to do so, one study by Thomson Reuters/FindLaw.com found half of adult Americans admit to using the Internet for personal use while at work.[7] This finding leads us to believe people are not very concerned about the monitoring of their personal data and hence have some level of trust that their employer has a strong enough ethical approach when it comes to capturing but not using this data in a malicious manner. The back and forth tug between employers and employees over privacy is nothing new.

Corporations are capturing enormous amounts of data. A study by Domo Inc - Data Never Sleeps 5.0, estimate 2.5 quintillion bytes of data are created every day.[8] The data captured by corporations can be divided into two different types; data collected internally from work being done by employees, and data collected on individuals who interact with the company, possibly as customers or in some other non-compensated manner. Corporations must have people and processes in place to deal with both types of data collected.

Corporations collect and keep data regarding vital statistics about their employees. Therefore, preventing network intrusions becomes a case of protecting employees' personal data. When a person is first hired for a job, much of their personal information is captured by the employer including sensitive information such as address, social security number, and phone number. If the employer provides health insurance, then they may collect information such as marital status, names and number of children in the employee's family and perhaps drug screening information. In the United States, information collected by an employer regarding their medical history must be kept confidential per the Americans with Disabilities Act (ADA), the Genetic

[7] "Half of Americans Use the Internet for Personal Reasons While at Work," 23 November 2015. [Online]. Available: https://www.thomsonreuters.com/en/press-releases/2015/november/americans-use-internet-personal-reasons-at-work-findlaw-survey.html. [Accessed 27 October 2018].

[8] "Data Never Sleeps 5.0," Domo Inc, 2017. [Online]. Available: https://www.domo.com/learn/data-never-sleeps-5?aid=ogsm072517_1&sf100871281=1. [Accessed 4 November 2018]

Information Nondiscrimination Act (GINA) and the Health Insurance Portability and Accountability Act (HIPAA).

Aside from this medical and disability information, companies have minimal legal obligations to protect employee data. Employees implicitly trust that their employer would keep personal data confidential. Examples like the Office of Personnel Management (OPM) data breach show the extent to which personal information can be released when due care is not taken in securing network infrastructure. In the case of the OPM breach, the records of 21.5 million current and former federal employees and contractors were exfiltrated.[9] People that were impacted by this breach had information from their SF-86 compromised, which includes previous employers, criminal history, family relationships, foreign contacts, and mental health history.[10,11]

8.2 Societal Impacts of Network Intrusions

The societal impact of network breaches is being felt as people lose control of their data. What is less evident is the ethical struggle that many companies are dealing with in disclosing that a breach occurred. It is not always in a company's best interest to disclose that a breach has occurred. That is why all 50 states including the District of Columbia, Guam, Puerto Rico, and the Virgin Islands have enacted some form of legislation requiring entities to notify individuals of security breaches where personally identifiable information is concerned[12]. Some industries have had this obligation to report breaches in a set amount of time, as is the case of healthcare providers with the HITECH Act [16], while other companies are under no such obligation. According to the Identity Theft Resource Center, in 2017, there were 1579 publicly disclosed breaches [17]. A different study done in 2018 by the Ponemon Institute found, on average, it takes organizations 197 days to even identify when a data breach has occurred and another 69 days to contain the breach [18].[13] Personal information can be stolen and used before anyone realizes the breach has occurred. The improper exposure of any of these pieces of data can cause personal harm. Most of the time hackers are attempting to find some way to achieve financial gain from stealing data or

[9] "Millions more Americans hit by government data hack." Reuters, 9 July 2015 [Online]. Available: https://www.reuters.com/article/us-cybersecurity-usa/millions-more-americans-hit-by-government-personnel-data-hack-idUSKCN0PJ2M420150709 [Accessed 4 November 2018]

[10] "OPM Hack Far Deeper Than Publicly Acknowledged, Went Undetected For More Than A Year, Sources Say", ABC News, 11 June 2015 [Online]. Available: https://abcnews.go.com/Politics/opm-hack-deeper-publicly-acknowledged-undetected-year-sources/story?id=31689059 [Accessed 4 November 2018]

[11] "Questionnaire for National Security Positions", OPM, 2010 [Online]. Available: https://www.opm.gov/forms/pdf_fill/sf86-non508.pdf [Accessed 4 November 2018]

[12] "Security Breach Notification Laws," National Conference of State Legislatures, 29 Sept 2018. [Online]. Available: http://www.ncsl.org/research/telecommunications-and-information-technology/security-breach-notification-laws.aspx.

[13] "2018 Cost of a Data Breach Study: Global Overview," Ponemon Institute LLC, 2018. [Online.] https://databreachcalculator.mybluemix.net/assets/2018_Global_Cost_of_a_Data_Breach_Report.pdf [Accessed 19 October 2018]

breaking into systems. We look at an example of these three types of breaches and the negative results and large numbers of people.

As corporations take additional steps to prevent or catch malicious activity against the organization, many costs are imposed on employees and the customers of the organization. Security controls are being imposed in many organizations, increasing the cost of doing business due to additional technology investments. The task of preventing intrusions becomes part of the employee's responsibility, heaping more administrative burden onto the workforce. Although this cost is shared by a number of people, many companies are still experiencing breaches in their security with varying impact on their business. While the cost to the business of implementing effective security controls can be significant, more penalties are being levied against companies that fail to protect against a data breach. With laws like HIPAA/HITECH, there are additional costs to bring a business' data standards up to the necessary level to ensure they are complying. Even though there can be benefits in updating the infrastructure to support the systems that comply with HITECH, i.e., electronic health records systems, the imposition of these requirements can take away from health care professionals providing adequate care instead of working toward maintaining compliance [19]. The goal is not to reduce positive outcomes with data protection; rather it is promoting an environment where adverse outcomes are less likely.

Personal health information has become a favorite target of hackers. Over the last 10 years, more and more health records have become electronic health records. The American Recovery and Reinvestment Act of 2009 incentivized medical professionals to adopt electronic health records.[14] This increase in availability means there is more opportunity for data breaches of health records. Since October of 2009, the number of individuals affected by stolen health records is a staggering 173,398,820 occurring across 1863 different breaches [20]. According to the "Health Warning" report by the Intel Security McAfee Labs, cybercriminals are putting more time and resources into exploiting and monetizing health care data.[15] It is clear that this type of data breach is a growing problem. Blackmail and extortion are the types of crimes that may go unreported. The reason for the extortion may be compromising information that individuals do not want to report to authorities. If criminals are successful in perpetrating this type of crime, then we can expect health records to continue to be a target of hackers as they find ways to monetize this data on the black market.

9 Conclusions

Our primary task was constructing features from a standard dataset made up of log data and identifying features that may be indicators of compromise which best detect

[14] "Medicare and Medicaid Health Information Technology: Title IV of the American Recovery And Reinvestment Act," 16 June 2009. [Online]. Available: https://www.cms.gov/Newsroom/MediaReleaseDatabase/Fact-sheets/2009-Fact-sheets-items/2009-06-16.html. [Accessed 06 October 2018].

[15] "Why data security is the biggest concern for health care," UIC Health Informatics, 2017. [Online.] https://healthinformatics.uic.edu/resources/articles/why-data-security-is-the-biggest-concern-of-health-care/ [Accessed 4 November 2018]

network intrusions. Using pre-planned controlled network intrusions to create red team logs is an effective method to analyze network behavior in an effort to identify specific indicators of compromise. Exploratory data analysis identified areas where features were created from raw data to come up with a model that was able to identify when IT professionals should examine network logs more closely for an intrusion. The features that we found that the top five indicators of compromise could be lumped into three major groups, new unique traffic, authentication events, and DNS queries. Using this methodology for identifying IOCs we were able to identify the constructed indicators that best predicted that an intrusion event happened.

While it is difficult to quantify the improvement, our model made on our initial requirements of this project, the framework used can be scaled up to accommodate the dataset and future development. While most IDS developers do not publish their response rates because the effectiveness will depend on the ruleset used, this model could be easily applied to work in an existing infrastructure to create additional layers of an existing security stance. Overtime with additional development the model used in this paper could provide better insight into the tools already used by a team to improve a group's chances of catching an intrusion.

10 Future Work

Our model provided some success in identifying data breaches given a training model on this particular type of network. This model should be tested against other data with similar logs to see if we attain similar performance. Our testing of classification algorithms was not exhaustive. Future work should include using other algorithms to see if better performance can be attained. Additional work would need to be in three primary areas, limitations with our ability to process that amount of data in the dataset, the ability to create dynamic features easily from the dataset, and the ability to remove periodicity from the created features.

Improved methods for finding outliers in the timeseries data would help for the classifiers identify significant features in the final model. Partially the issue was the lack of a suitable period to create a baseline for normal use, and the other is the chosen library used to identify outliers in the data. While the Prophet library is good at fitting a normalized model to the data, it also introduced unwanted periodicity into some features like data that was extracted from the DNS logs. Ideally, an improved algorithm would be used to train on a baseline period and then used to identify outliers in the data with actual red-team events.

References

[1] K. Scarfone, M. Souppaya, A. Cody and A. Orebaugh, *Technical Guide to Information Secuirty Testing and Assessment,* Gaithersburg: NIST, 2008.

[2] K. Scarfone and P. Mell, *Guide to Intrusion Detection and Prevention System (IDPS),* Gaithersburg MD: National Institute of Standards and Technology, 2007.

[3] H. T. Nguyen, K. Franke and S. Petrović, Feature Extraction Methods for Intrusion Detection Systems, 2012.

[4] W. Heyi, H. Aiqun, S. Yubo, B. Ning and J. Xuefei, "A new intrusion detection feature extraction method based on complex network theory," in *2012 Fourth International Conference on Multimedia Information Networking and Security*, 2012.

[5] L. Xie and J. Li, "A Novel Feature Extraction Method Assembled with PCA and ICA for Network Intrusion Detection," in *2009 International Forum on Computer Science-Technology and Applications*, 2009.

[6] F. Wang, S. Wang, Y. Bai and W. Che, "Feature Extraction Method for Network Intrusion Detection Based on RS-KPCA," *Applied Mechanics and Materials,* pp. 706-711, October 2014.

[7] T. M. Pattewar and H. A. Sonawane, "Neural network based intrusion detection using Bayesian with PCA and KPCA feature extraction," in *2015 IEEE International Conference on Computer Graphics, Vision and Information Security*, 2015.

[8] H. Mannila, H. Toivonen and A. Inkeri Verkamo, "Discovery of Frequent Episodes in Event Sequences," *Data Mining and Knowledge Discovery,* pp. 259-389, 1997.

[9] A. Jain, R. Duin and J. Mao, "Statistical pattern recognition: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 1, pp. 4-37, 2000.

[10] W. Lee, S. J. Stolfo and K. W. Mok, "A Data Mining Framework for Building Intrusion Detection Models," in *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, New York, NY, 1999.

[11] E. M. Hutchins, M. J. cloppert and R. M. Amin, "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains," in *Proceedings of the 6th International Conference on Information Warfare and Security*, Washington DC, 2010.

[12] A. D. Kent, *Comprehensive, Multi-Source Cyber-Security Events,* Los Alamos National Laboratory, 2015.

[13] W. Lee, *A data mining framework for constructing features and models for intrusion detection systems,* New York, NY: ProQuest Dissertations Publishing, 1999.

[14] S. D. Warren and L. D. Brandeis, "The Right to Privacy," *Harvard Law Review,* vol. 4, no. 5, pp. 193-220, 1890.

[15] M. C. Calisti, "You Are Being Watched: The Need for Notice in Employer Electronic Monitoring," *Kentucky Law Journal,* vol. 96, p. 649, 2007.

[16] Department of Health and Human Services, *HIPAA Administrative Simplification: Enforcement,* Federal Register, 2009.

[17] Identity Theft Resource Center, "2017 Annual Data Breach Year-End Review," Identity Theft Resource Center, 2018.

[18] Ponemon Institute LLC, "2017 Cost of Data Breach Study," Ponemon Institute, Traverse City, MI, 2017.

[19] J. D. Halamka and M. Tripathi, "The HITECH Era in Retrospect," *The New England Journal of Medicine,* vol. 377, no. 10, pp. 907-909, 2017.

[20] W. Koczkodaj, M. Mazurek, D. Strzałka, A. Wolny-Dominiak and M. Woodbury-Smith, "Electronic Health Record Breaches as Social Indicators," *Social Indicators Research,* pp. 1-11, Feb 2018.