Anomaly Detection in Host Log Files

Kate Stadelman

Executive Summary

In today's digital age, cybersecurity has become a primary concern for business. While threats come in many forms, compromised credentials continue to be a main cause of costly breaches. It is essential that harmful user activity is identified and dealt with as quickly as possible, and this has become a fundamental capability of many cyber protection solutions. In this paper, we research whether we can identify anomalous user activity in massive computer (host) log files. Using Isolation Forest, we show that not only can we identify potentially compromised users, but we can do so with highly irregular and limited training data.

Introduction

Even amid COVID-19, cybersecurity remains a primary focus for businesses. "71% of US CEOs said they are 'extremely concerned' about cyber threats – ahead of pandemics and other health crises (46%)" (PwC, 2021). These concerns are warranted given the high cost associated with cyberattacks. According to IBM's most recent Cost of a Breach Report, "compromised credentials were responsible for 20% of breaches at an average breach cost of USD $4.37 million" (IBM, 2021). Once low-level user access is obtained, bad actors typically attempt to escalate privileges to administrative levels. With high-level access, bad actors may hijack processes, steal information, destroy data, or pivot their attack, compromising another host on the network. Thus, it is of the utmost importance for organizations to track user activity, including login attempts and failures, privileged actions, computers (host) accessed, and processes run. This data can then be used for analysis and identification of potentially harmful user behavior patterns, leading us to our research question:

Can we detect suspicious user activity hidden in massive computer (host) log files?

Data Source & Definitions

Unified Host and Network Data Set

Cybersecurity data sets are notoriously difficult to obtain because there is a risk of disclosing too much information, leaving authoring organizations vulnerable to future cyberattacks. Thankfully, Los Alamos National Laboratory created the Unified Host and Network Data Set, “a subset of network and computer (host) events collected [. . .] over the course of approximately 90 days” (Turcotte et al., 2018), which they deidentified to protect their network. This research project utilizes hosts logs from Day 1, totaling 55.6 million records. Detailed descriptions of data set elements may be found in Tables 1-2, but generally, these logs track login, process, and privileged events. Certain events, such as workstations locking / unlocking and screensavers starting / stopping, were irrelevant to the research question and therefore excluded.

Log Aggregation

Tracing events back to the originating username became particularly important for log aggregation, which was accomplished using the Time, LogHost, UserName, LogonID, SubjectUserName, and SubjectLogonID fields. Parent usernames were determined using the minimum event Time for each unique LogHost and LogonID combination, or LogHost and SubjectLogonID combination if SubjectLogonID was populated. Logs were aggregated by parent username to produce the following measures (as described in Table 3): number of logins attempted, login failures, computers (hosts) accessed, processes started, privileged actions, and mapped credentials. The resulting aggregated data set contained 19,220 observations.

Exploratory Data Analysis

Initial analysis of the aggregated data set found measures to have highly skewed distributions with extreme scales. To address concerns that a small number of outliers would unduly impact our model, log transformations were applied. Figures 1-6 display qqplots, box plots, and histograms of our transformed measures, which show that our data is still not normally distributed and contains a fair number of outliers. Fortunately, our methodology, Isolation Forest, does not have strict distribution requirements and performs well on non-normal data.

Additionally, we prepared a scatterplot matrix with Pearson Correlation Coefficients (Figure 7) to determine if any features were redundant. While we cannot assume that each observation is independent because more than one username might be compromised, or at the very least be used by the same person, the Person Correlation Coefficient of 0.841 for Login Attempts and Mapped Credentials implies that Mapped Credentials are a proxy for Login Attempts. This proved out under further examination, as Mapped Credentials were found to contain a high number of Windows Active Directory (Kerberos) events, which are essentially pass-through login activities. Consequently, we excluded Mapped Credentials from our model.

Isolation Forest

Initially proposed in 2008 by Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, Isolation Forest is the first anomaly detection methodology that seeks to explicitly isolate anomalies, rather than generating profiles of “normal” data prior to identifying outliers (Liu et al., 2008). Due to this fundamental shift in approach, Isolation Forest is an exceptionally performant algorithm with low memory requirements and linear time complexity (Young, 2020). In addition to deftly handling sizable data sets, Isolation Forest is broadly known to produce accurate results even with suboptimal training data. Isolation Forest has been “effectively trained even in the absence of anomalies in the sample,” as well as when training data include a large number of irrelevant features (Zhu & Suresh, 2019). Moreover, there are no requirements relating to data distribution or normality.

An extension of Random Forest, Isolation Forest is an unsupervised, tree-based methodology based on the notion that “since anomalies are ‘few and different’ [. . .] they are more susceptible to isolation” (Liu et al., 2008). Utilizing random partitioning, Isolation Forest generates an ensemble of binary isolation trees:

> In a data-induced random tree, partitioning of instances are repeated recursively until all instances are isolated. This random partitioning produces noticeable shorter paths for

> anomalies since (a) the fewer instances of anomalies result in a smaller number of partitions – shorter paths in a tree structure, and (b) instances with distinguishable attribute-values are more likely to be separated in early partitioning. Hence, when a forest of random trees collectively produce shorter path lengths for some particular points, then they are highly likely to be anomalies. (Liu et al., 2008)

Where Random Forest seeks to predict an observation's value or classification based on the majority rule of multiple decision trees, Isolation Forest assigns observations an anomaly score based on the average path length to each node (Zhu & Suresh, 2019).

Because isolation trees have an "equivalent structure" to Binary Search Tree (BST), the average path length of unsuccessful search in BST is used as a foundation for how Isolation Forest estimates the average path length to each node (Liu et al., 2008). Given a sample of *n* observations, BST calculates the average path length of unsuccessful search c as:

$$c(n) = 2H(n-1) - (2(n-1)/n)$$

where $H(i)$ is a "harmonic number [. . .] estimated by $\ln(i) + 0.5772156649$ (Euler's constant)" (Liu et al., 2008). In Isolation Forest, given a sample of *n* observations, the anomaly score *s* of observation *x* is:

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$$

where $h(x)$ is the path length of *x* (number of edges from tree root to node), $E(h(x))$ is the "average of $h(x)$ from a collection of isolation trees," and the normalizing factor $c(n)$ is defined as in BST above (Liu et al., 2008).

Anomaly scores range in value between 0 and 1. If an observation has an anomaly score smaller than 0.5, it can be safely considered normal. Alternatively, observations with anomaly scores closer to 1 are highly likely to be anomalous. If all observations have anomaly scores close to 0.5, then the entire sample may be regarded as normal (Liu et al., 2008).

Model Implementation

‘isotree’ Package

For our Isolation Forest model, we applied the ‘isotree’ R package. Described by its authors as a “fast and multi-threaded implementation of isolation forest” (Cortes et al., 2021), ‘isotree’ offers many tuning parameters (highlighted in Table 4).. Package documentation details parameter settings that mimic previous Isolation Forest implementations, namely the original ‘iForest’ and subsequent ‘EIF’ and ‘SCiForest’ algorithms. However, the authors of ‘isotree’ indicate that default parameters are “more likely to result in better models” (Cortes et al., 2021). Given the successful outcome of our model, we did not find cause to change ‘isotree’ default parameter settings.

Training & Test Data

In early stages of our research, we randomly split our aggregated data into training (70%) and test (30%) sets. However, because Isolation Forest is known to correctly identify anomalies even with small training samples, we iteratively reduced the proportion of our training set by 10% until we observed a degradation in model outcomes. Ultimately, we split training and test sets 50% / 50%, which resulted in a similar distribution of anomaly scores as the original 70% / 30% split. These final training and test data sets contained 9,610 observations each.

Model Training

After training our model, we plotted the density of outputted anomaly scores (Figure 8), finding the majority of observations were considered normal (anomaly score ≤ 0.5). To highlight the most anomalous users and visually review the accuracy of our model, we set an outlier threshold of anomaly score ≥ 0.7 and created a scatterplot matrix with anomalous observations in coral and normal observations in teal (Figure 9). Unsurprisingly, detected anomalies were often, but not always, observations with the largest value for a given measure, as seen in the variable distributions plotted in the diagonal of Figure 9.

Results

Pleased with the training results, we moved forward with applying our model to the test data. Using the predicted anomaly scores and previously determined outlier threshold, we produced a density plot and scatterplot matrix of test results (Figures 10-11), which appeared very consistent with our plots from model training. Additionally, we reviewed specific users identified as anomalies (Table 5). Certain users, including Comp065845$, User515356, and User031784, were clearly worrisome with a large number of login failures, while others, such as User643724 and User324202, had no login failures but an unusually high number of privileged actions and hosts accessed, respectively. While we cannot fully verify if users were compromised without access to the original, non-deidentified logs, all users identified by our model did indeed appear worthy of review for suspicious activity.

Conclusion

As business continues to navigate the perils of the digital age, we must also continue to purse cyber protection solutions that prevent cyberattacks and safeguard data. In this research paper, we sought to detect suspicious user activity in massive computer (host) log files. Aggregating logs from the Unified Host and Network Data Set, we successfully applied Isolation Forest, generating a list of users who displayed anomalous behavior.

References

Cortes, D., Goetghebuer-Planchon, T., Blackman, D., & Vigna, S. (2021, October 23). Isolation-Based Outlier Detection ('isotree') [R Package]. https://cran.r-project.org/web/packages/isotree/isotree.pdf

International Business Machines Corporation (IBM). (2021). Cost of a Data Breach Report 2021. https://www.ibm.com/security/data-breach

Liu F. T., Ting K. M., & Zhou, Z.-H. (2008). Isolation Forest. *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM'08), Pisa, Italy*. https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/icdm08b.pdf

PricewaterhouseCoopers (PwC). (2021). PwC 24th Annual Global CEO Survey. https://www.pwc.com/gx/en/ceoagenda/ceosurvey/2021.html

Turcotte M., Kent A., & Hash C. (2018, November). Unified Host and Network Data Set. *Data Science for Cyber-Security*, (1–22). https://www.worldscientific.com/doi/abs/10.1142/9781786345646_001

Young, A. (2020, November 13). Isolation Forest is the best Anomaly Detection Algorithm for Big Data Right Now. *Towards Data Science*. https://towardsdatascience.com/isolation-forest-is-the-best-anomaly-detection-algorithm-for-big-data-right-now-e1a18ec0f94f

Zhu, A. & Suresh, S. (2019, December 13). Isolation Forest for Data Mining. *Medium*. https://medium.com/@siddharth.suresh92/isolation-forest-for-data-mining-a2c44a26d646

Table 1

Host Log Data File Elements

Element	Element Description
Time	The epoch time of the event in seconds.
EventID	Four-digit integer corresponding to the event id of the record.
LogHost	The hostname of the computer that the event was recorded on. In the case of directed authentication events, the Log Host will correspond to the computer that the authentication event is terminating at (destination computer).
LogonType	Integer corresponding to the type of logon.
LogonTypeDescription	Description of the Logon Type.
UserName	The user account initiating the event. If the user ends in $, then it corresponds to a computer account for the specified computer.
DomainName	Domain name of UserName.
LogonID	A semi-unique (unique between current sessions and LogHost) number that identifies the logon session just initiated. Any events logged subsequently during this logon session should report the same Logon ID through to the logoff event.
SubjectUserName	For authentication mapping events, the user account specified by this field is mapping to the user account in UserName.
SubjectDomainName	Domain name of SubjectUserName.
SubjectLogonID	See LogonID.
Status	Status of the authentication request. “0x0” means success otherwise failure.
Source	For authentication events, this will correspond to the the computer where the authentication originated (source computer), if it is a local logon event then this will be the same as the LogHost.
Service Name	The account name of the computer or service the user is requesting the ticket for.
Destination	This is the server the mapped credential is accessing. This may indicate the local computer when starting another process with new account credentials on a local computer.
AuthenticationPackage	The type of authentication occurring including Negotiate, Kerberos, NTLM plus a few more.
FailureReason	The reason for a failed logon.

ProcessName	The process executable name, for authentication events this is the process that processed the authentication event. ProcessNames may include the file type extensions (i.e. exe).
ProcessID	A semi-unique (unique between currently running processes AND LogHost) value that identifies the process. ProcessID allows you to correlate other events logged in association with the same process through to the process end.

Note. Reprinted from "Unified Host and Network Data Set" (Turcotte et al., 2018).

Table 2

Host Log Event Descriptions by Event ID

Event ID	Event Description
4768	Kerberos authentication ticket was requested (TGT)
4769	Kerberos service ticket was requested (TGS)
4770	Kerberos service ticket was renewed
4774	An account was mapped for logon
4776	The domain controller attempted to validate the credentials for an account
4624	An account was successfully logged on
4625	An account failed to logon on
4634	An account was logged on
4647	User initiated logon
4648	A logon was attempted using explicit credentials
4672	Special privileges assigned to a new logon
4800	The workstation was locked
4801	The workstation was unlocked
4802	The screensaver was invoked
4803	The screensaver was dismissed
4688	Process start
4689	Process end
4608	Windows is starting up
4609	Windows is shutting down

1100	Event logging service has shut down (often recorded instead of EventID 4609)

Note. Reprinted from "Unified Host and Network Data Set" (Turcotte et al., 2018).

Table 3

Username Measures

Measure	Description
Username	Parent username who instigated activity
Login Attempts	Number of login and authorization attempts
Login Failures	Number of login failures
Hosts Accessed	Number of computers (hosts) accessed
Processes Started	Number of processes started (code executed)
Privileged Actions	Number of logins and processes started as Administrator
Mapped Credentials	Number of unique usernames accessed by parent username

Note. Aggregations calculated by parent username.

Table 4

Selected 'isotree' Parameters

Parameter	Description
data	Data to which to fit the model. Supported inputs type are: • A 'data.frame', also accepted as 'data.table' or 'tibble'. • A 'matrix' object from base R. • A sparse matrix in CSC format, either from package 'Matrix' (class 'dgCMatrix') or from package 'SparseM' (class 'matrix.csc').
sample_size	Sample size of the data sub-samples with which each binary tree will be built. Recommended value in references is 256, while the default value in the author's code is 'nrow(data)'.
ntrees	Number of binary trees to build for the model. Recommended value in reference is 100, while the default value in the author's code is 10.
ndim	Number of columns to combine to produce a split.
ntry	In the extended model with non-random splits, how many random combinations to try for determining the best gain.

categ_cols	Columns that hold categorical features, when the data is passed as a matrix (either dense or sparse). Can be passed as an integer vector (numeration starting at 1) denoting the indices of the columns that are categorical, or as a character vector denoting the names of the columns that are categorical, assuming that 'data' has column names.
max_depth	Maximum depth of the binary trees to grow. By default, will limit it to the corresponding depth of a balanced binary tree with number of terminal nodes corresponding to the sub-sample size (the reason being that, if trying to detect outliers, an outlier will only be so if it turns out to be isolated with shorter average depth than usual, which corresponds to a balanced tree depth).
output_score	Whether to output outlierness scores for the input data, which will be calculated as the model is being fit and it's thus faster. Cannot be done when using subsamples of the data for each tree (in such case will later need to call the 'predict' function on the same data).
nthreads	Number of parallel threads to use. If passing a negative number, will use the maximum number of available threads in the system.

Note. Excerpted from "Isolation-Based Outlier Detection" (Cortes et al., 2021).

Table 5

Anomalous Users Identified by Isolation Forest

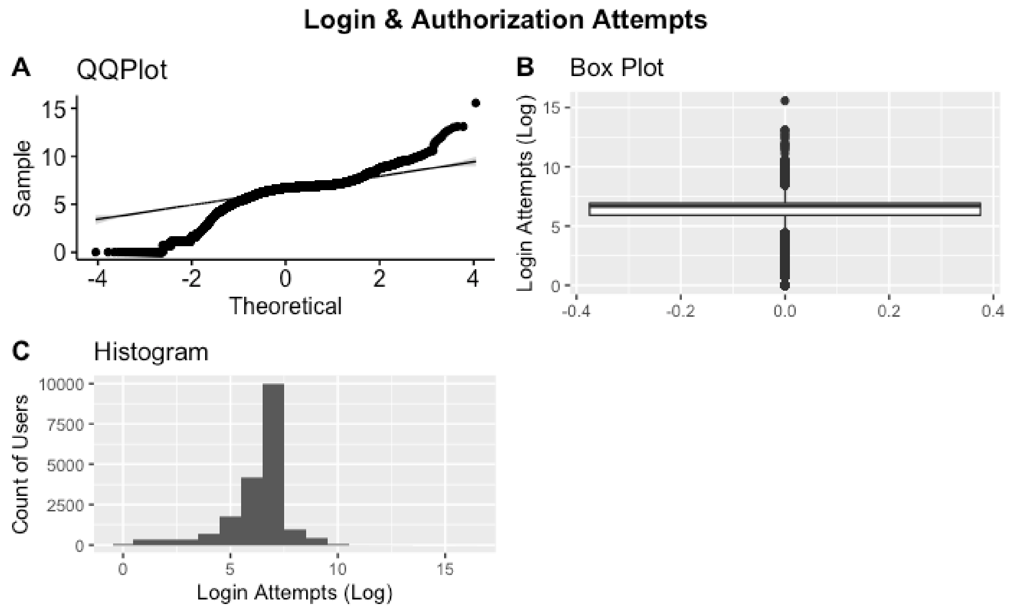
User Name	Anomaly Score	Login Attempts	Login Failures	Hosts Accessed	Processes Started	Privileged Actions	Mapped Credentials
Comp065845$	0.777	153,964	153,131	2	831	21	438
User006226	0.747	10,514	5,210	11	752	0	10,425
User515356	0.747	34,728	17,264	4	42	2	34,659
User031784	0.747	39,844	19,595	8	39	3	39,711
User816098	0.744	14,905	2,875	4	0	0	6,031
User071989	0.739	6,121	2,162	7	396	561	2,249
User587067	0.735	14,939	2,873	5	5	0	6,025
User096590	0.735	14,943	2,873	5	12	10	6,057
User829284	0.730	17,382	1,917	11	66	0	8,602
User643724	0.728	493,756	0	3	160	108,994	246,641
User324202	0.716	583	0	90	1	196	0

User247683	0.714	3,098	778	2	0	0	778
User015659	0.704	5,970	2,218	8	78	1	956

Note. While we cannot determine if users are conducting harmful activity without access to the original (non-deidentified) logs, all users identified as anomalous by Isolation Forest appear to have abnormal user activity.

Figure 1

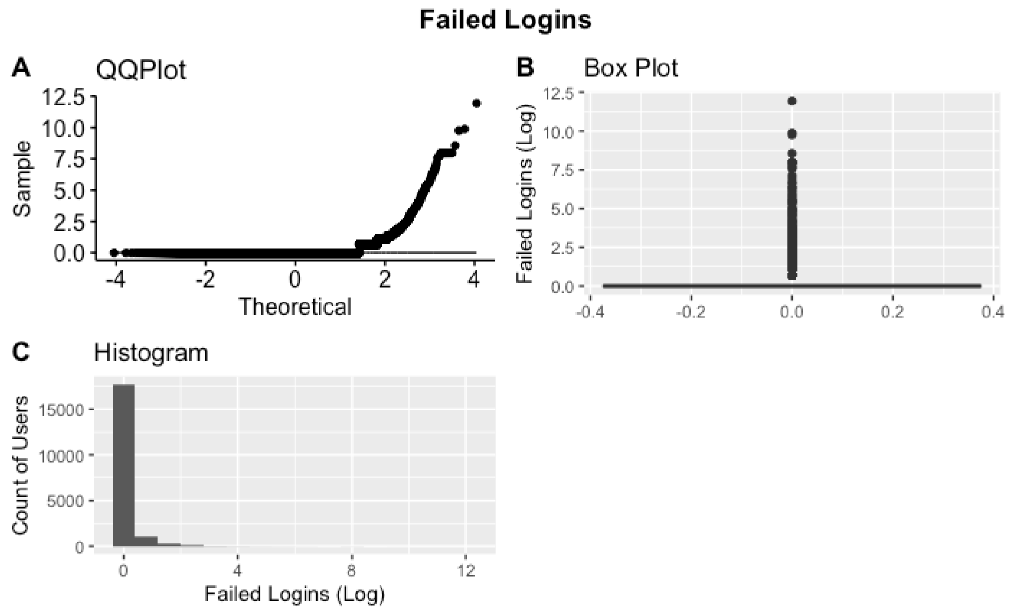
QQPlot, Box Plot, and Histogram Depicting Login and Authorization Attempts Per User



Note. A log transformation was applied to the measure due to highly skewed data.

Figure 2

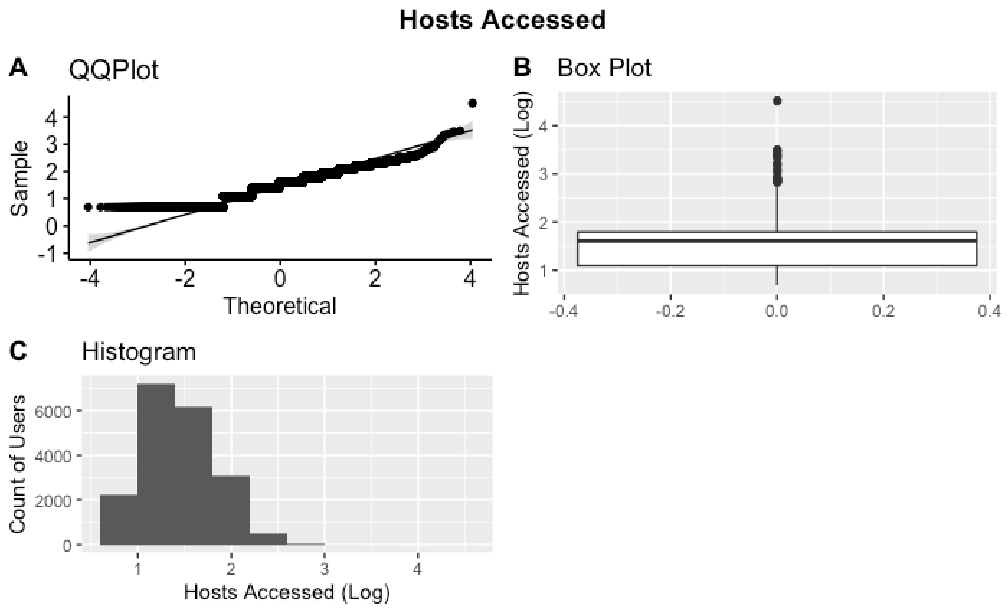
QQPlot, Box Plot, and Histogram Depicting Login Failures Per User



Note. A log transformation was applied to the measure due to highly skewed data.

Figure 3

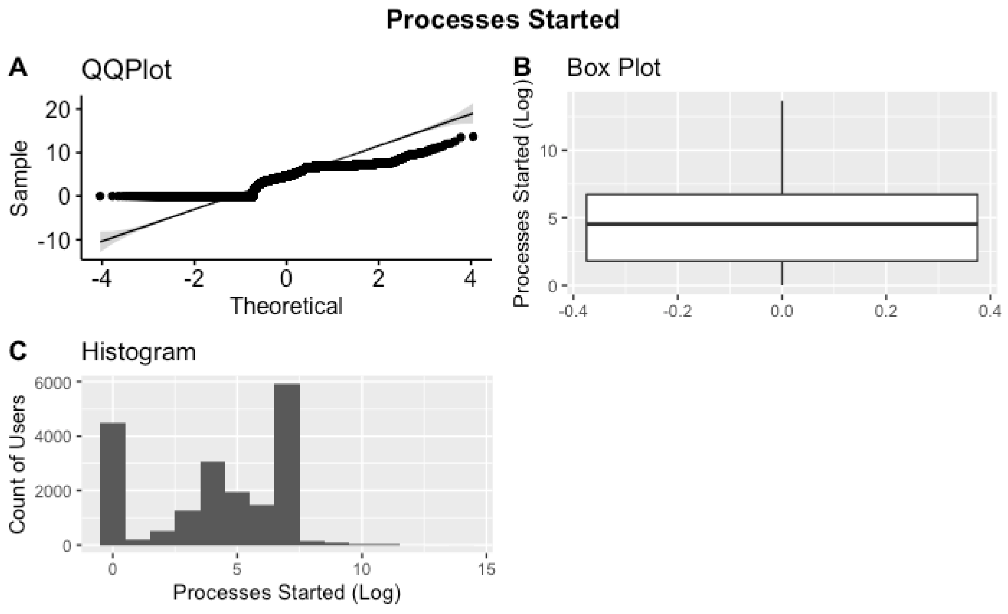
QQPlot, Box Plot, and Histogram Depicting Computers (Hosts) Accessed Per User



Note. A log transformation was applied to the measure due to highly skewed data.

Figure 4

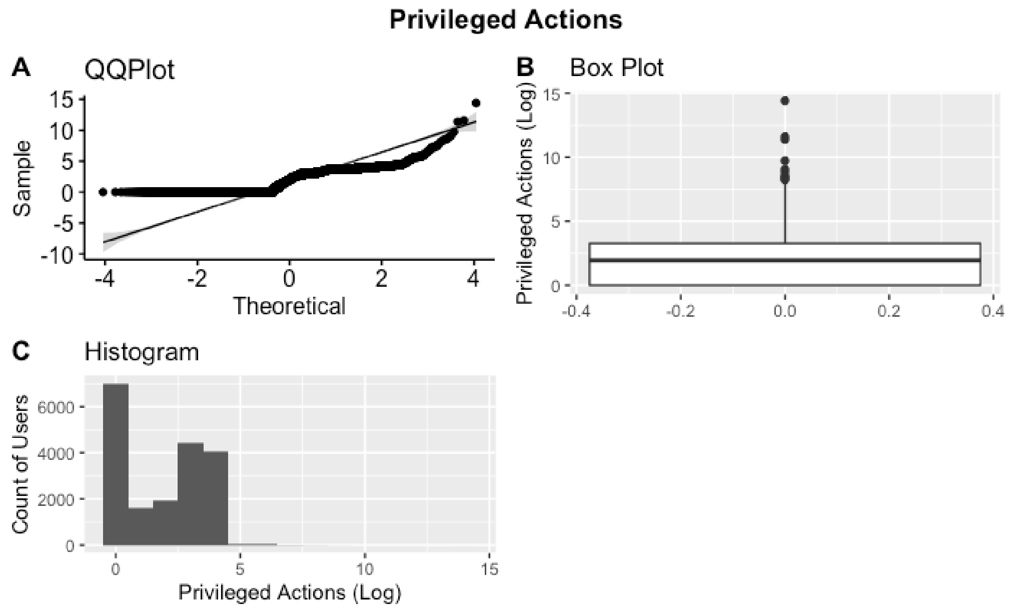
QQPlot, Box Plot, and Histogram Depicting Processes Started Per User



Note. A log transformation was applied to the measure due to highly skewed data.

Figure 5

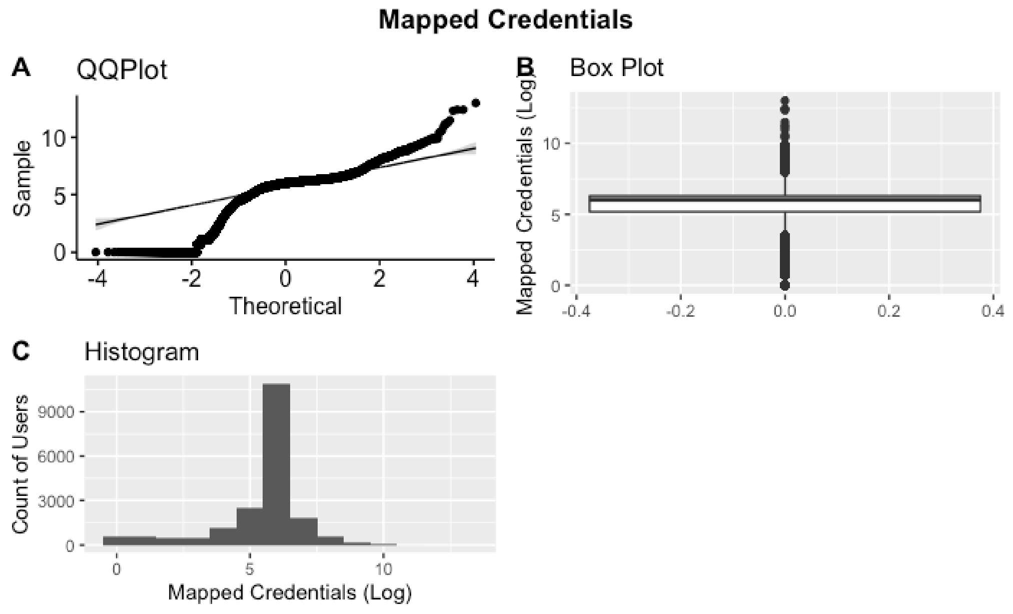
QQPlot, Box Plot, and Histogram Depicting Privileged Actions Per User



Note. A log transformation was applied to the measure due to highly skewed data.

Figure 6

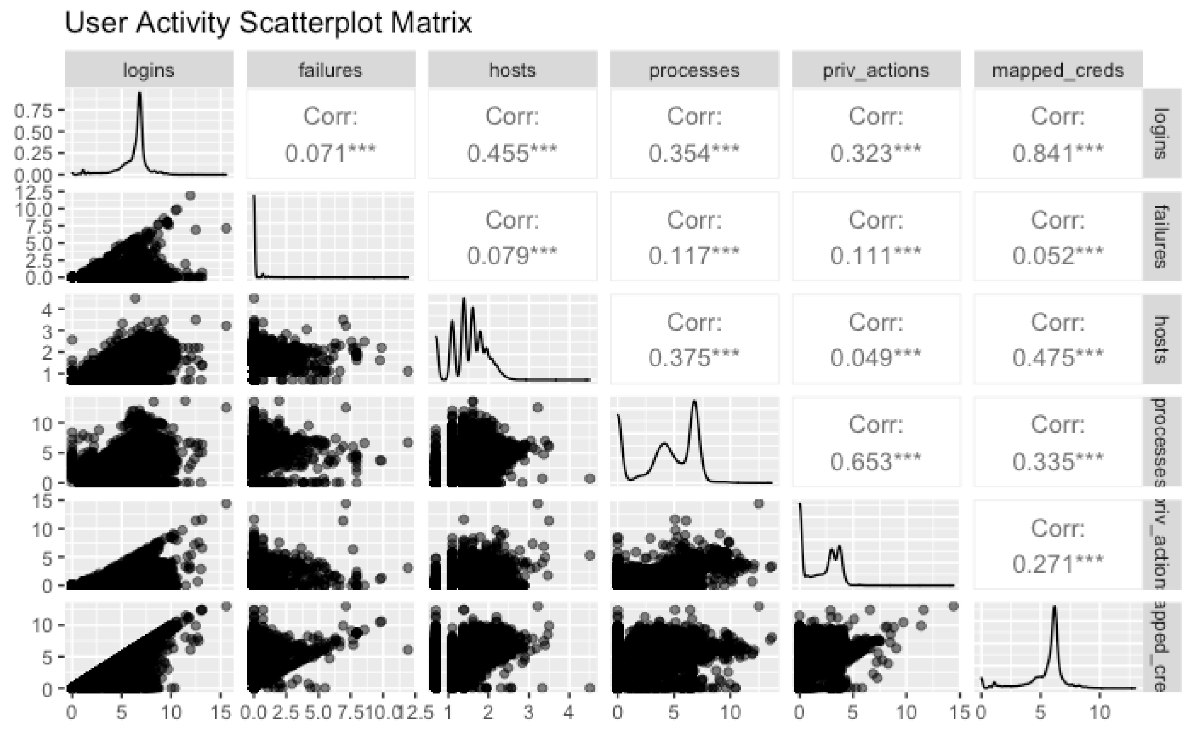
QQPlot, Box Plot, and Histogram Depicting Mapped Credentials Per User



Note. A log transformation was applied to the measure due to highly skewed data.

Figure 7

Scatterplot Matrix of Username Features with Pearson Correlation Coefficients



Note. We cannot assume that username activity is independent because more than one username may be compromised by the same bad actor. However, we can use this information to assess if any features are redundant. From this visualization, Mapped Credentials appear redundant to Login Attempts. Upon review, we found a significant proportion of Mapped Credential logs were Windows Active Directory (Kerberos) events, which are essentially pass-through login attempts.

Figure 8

Anomaly Score Density Plot for Training Data Set

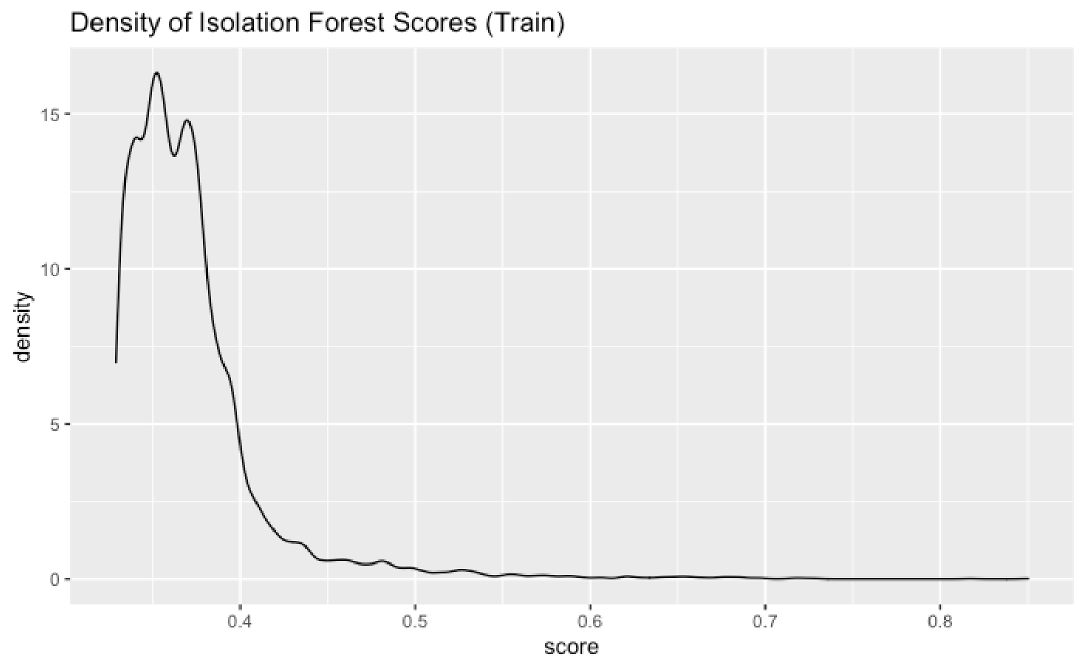


Figure 9

Scatterplot Matrix of Username Features Designating Outlier vs. Normal Activity for Training Data Set

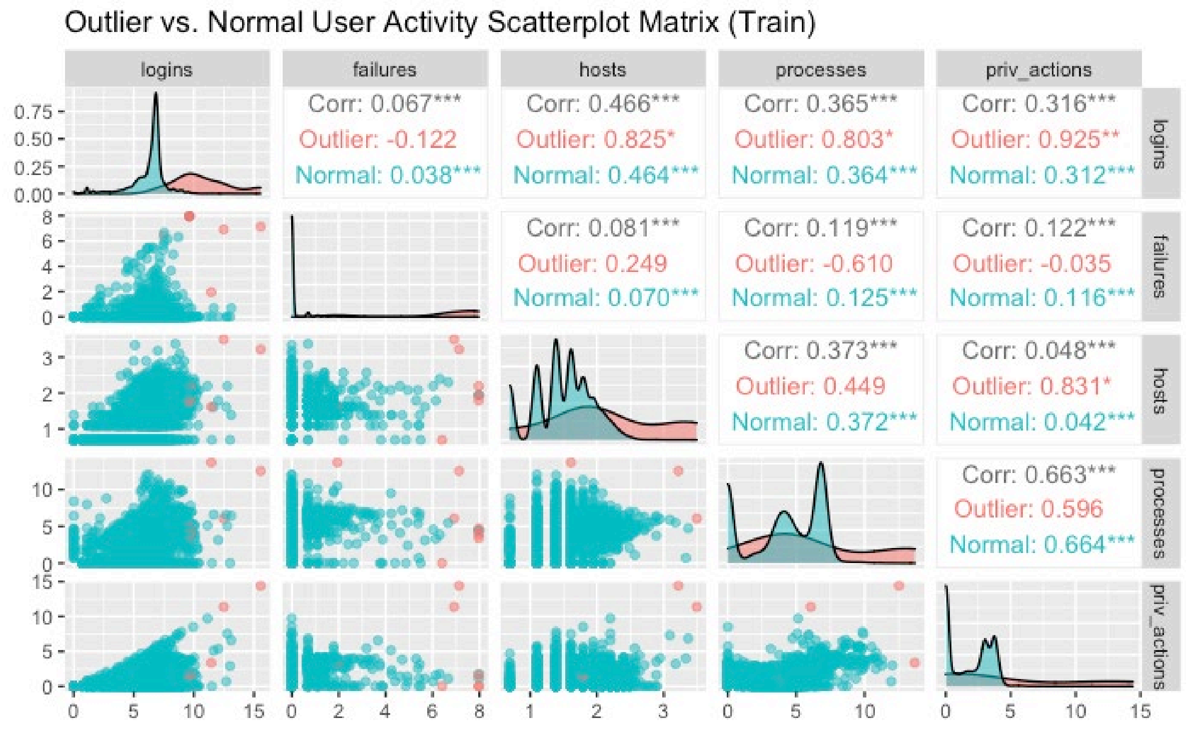


Figure 10

Anomaly Score Density Plot for Test Data Set

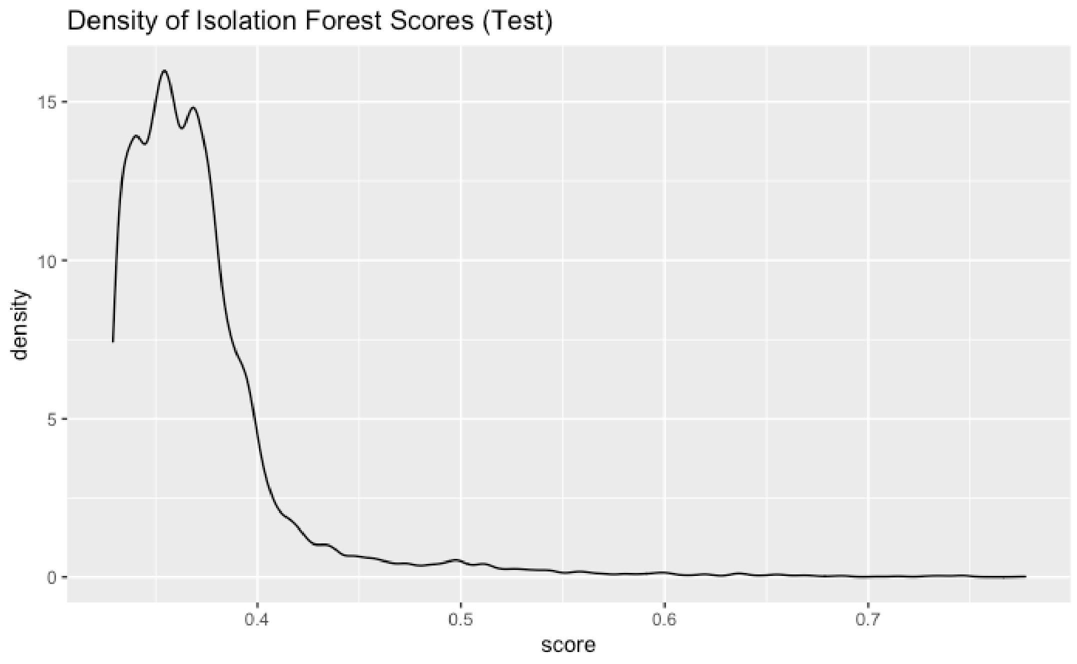
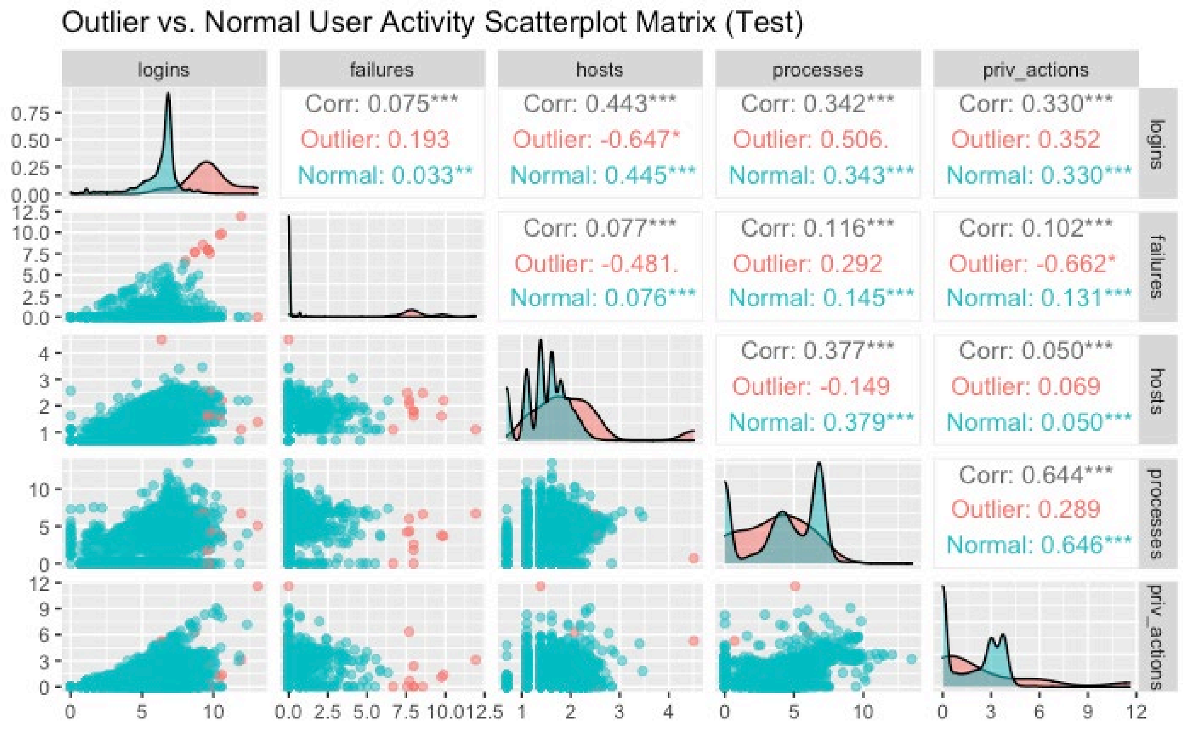


Figure 11

Scatterplot Matrix of Username Features Designating Outlier vs. Normal Activity for Test Data Set


Outlier vs. Normal User Activity Scatterplot Matrix (Test)
logins
failures
hosts
processes
priv_actions
Corr: 0.075***
Outlier: 0.193
Normal: 0.033**
Corr: 0.443***
Outlier: -0.647*
Normal: 0.445***
Corr: 0.342***
Outlier: 0.506.
Normal: 0.343***
Corr: 0.330***
Outlier: 0.352
Normal: 0.330***
Corr: 0.077***
Outlier: -0.481.
Normal: 0.076***
Corr: 0.116***
Outlier: 0.292
Normal: 0.145***
Corr: 0.102***
Outlier: -0.662*
Normal: 0.131***
Corr: 0.377***
Outlier: -0.149
Normal: 0.379***
Corr: 0.050***
Outlier: 0.069
Normal: 0.050***
Corr: 0.644***
Outlier: 0.289
Normal: 0.646***