Agglomerative Hierarchical Clustering as an Unsupervised Anomaly Detection Technique

Network Intrusion Detection via Zeek Log Analysis

Sean P. Murphy

19 July 2022

Contents

Part I: Research Question

Can an agglomerative hierarchical clustering analysis of network Zeek logs function as an effective anomaly detection technique?

Context

Network Intrusion Detection Systems (NIDS) are installed as physical devices and/or software applications that monitor network traffic patterns to identify malicious activity so that system administrators can take appropriate actions to ensure the integrity and security of their network (Newton & Schoen, 2022). The two primary methods that NIDS use to classify threats are signature libraries and anomaly detection.

Signature libraries are static definition lists that contain digital fingerprints of known exploits; the NIDS identifies and flags network traffic matching the pattern of any threat definition in the signature library. This method is valid against known exploits, also known as n-day exploits, because the NIDS can effectively label the traffic as malicious and give the systems administrator an indication of how the compromise occurred and what the intended effect may have been. However, signature libraries are completely ineffective with respect to zero-day exploits; these attacks are either entirely unknown to cybersecurity specialists or are too new to have been fingerprinted sufficiently well to develop a signature-based detection definition. For these threats, an anomaly-based methodology is more effective.

Anomaly-based detection methods measure network activity against an established baseline of normal and authorized traffic patterns; any network or user behavior that does not fall within the scope of the baseline is flagged as suspicious and—therefore—a potential indicator of compromise. These methods are equally valid approaches against either n-day exploits or zero-day exploits.

Justification

It is sometimes necessary to assess the security status of a network that does not have a NIDS installed or after a period of time during which an installed NIDS was inactive. In this case, without a system monitoring traffic in real time, a packet capture (PCAP) or alternative network logging system can be leveraged. While PCAP data is the gold standard for network forensics, the effort and expense required to collect and store every packet of network traffic in its entirety limits this kind of application to very narrow and temporary use cases (Sikos, 2020). The ability to use a more common and accessible logging standard such as Zeek as an input dataset for an anomaly detection algorithm would be an option that carries significant business value with respect to information assurance, audit compliance, and risk management assessments (Andrews et al., 2019).

The contribution of this study to the field of Data Analytics and the MSDA program is to create a clustering model which can identify anomalous network activity from Zeek log data for further investigation by cybersecurity analysts so that potential threats are flagged for remediation before causing significant damage to compromised information systems (The Zeek Project, 2020). This study will use agglomerative hierarchical clustering, an unsupervised machine learning method for data mining, to leverage multiple features of stored Zeek logs in order to create groupings of data points based on their relative similarity in order to isolate any anomalous network activity that merits additional investigation. According to Mazarbhuiya (2019), agglomerative hierarchical clustering is an ideal method to employ for identifying anomalies in the collective attributes of network traffic data which serves as an effective intrusion detection mechanism.

Hypotheses

This study will assume the status quo of a network without a functioning NIDS and no PCAP data available. Namely, the null hypothesis will be that this analysis will not produce any usable information due to the

lack of a dedicated monitoring system or full packet capture and storage. Hence, the alternative hypothesis will be that a clustering analysis of Zeek logs can serve a similar function as a NIDS employing anomaly detection by indicating outliers that merit additional scrutiny from a cybersecurity analyst.

H_0: A hierarchical clustering analysis of Zeek logs cannot reveal anomalous network behavior.

H_A: A hierarchical clustering analysis of Zeek logs can reveal anomalous network behavior.

Part II: Data Collection

Collection Process

The data needed to address the above defined research question should be Zeek log data collected from a network with a non-trivial number of active users, multiple public-facing hosted services, and—ideally—was compromised during the log collection period. These attributes ensure that the sample data incorporates multiple types of network traffic and will allow a complete assessment of whether the above defined null hypothesis is accepted or rejected (Zhang, 2010).

The U.S. Army Cyber Command's (ARCYBER) Information Integration Division (ID2) maintains Zeek logs for a subset of Department of Defense Information Network (DODIN) systems. Some of this data has been approved for release to defense industry and academic partners for research and development purposes (Brust, 2021). With that purpose in mind, the ARCYBER Cyber Protection Brigade (CPB) identifies candidate sample data sets while conducting their primary mission of identifying and mitigating real-world network intrusions on real-world networks.

Collection Assessment

This data collection process has the inherent advantage of ensuring the requisite data complexity for this project—including the presence of malicious traffic resulting from that network's compromise. Conversely, one disadvantage of using this technique for data collection is that every aspect of the collection process occurs under the control of a party not directly or indirectly involved with this study. This means that any desired adjustments to the quantity of data (e.g., date and time ranges available to query) are impossible. This also means that feature selection for this project will be driven by the Zeek configuration at the site of network compromise as opposed to a custom Zeek configuration optimized for anomaly detection.

Part III: Data Extraction and Preparation

Data Extraction

Access to this data set was secured via a formal request through ID2, coordinating with Gabriel Nimbus (GN) administrators to extract the data from its Hadoop cluster, and arranging the transfer of the extracted data to a DODIN-external server for analysis.

Data was queried via the GN portal using a temporary purpose-built account with minimal permissions to prevent access to any sensitive or protected data. As such, the query "*" was run across the widest date range authorized in order to ensure that all available data was collected. This data was extracted from the GN Hadoop cluster as a comma separated values (.csv) file and transferred to a private server so that it could be accessed from a non-DODIN system.

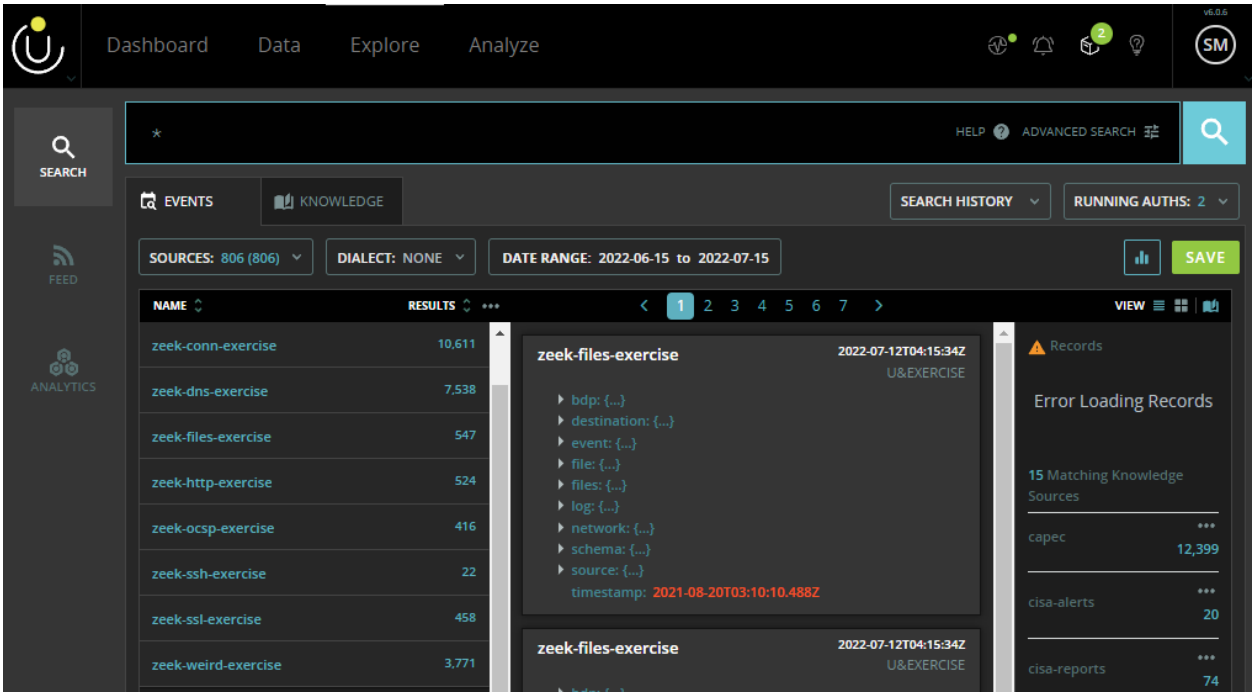

Figure 1: Screenshot of Gabriel Nimbus User Interface

Data Preparation

While it is true that multiple tools are capable of performing agglomerative hierarchical clustering, R will be employed for this task due to its greater diversity of packages that are more specifically targeted to narrow applications as compared to Python or SAS (Keells, 2021). Due to being created by statisticians for statistical and data science purposes, R also performs many integral tasks natively and—hence—more simply and quickly than alternative tools (Wickham & Grolemund, 2017).

The extracted comma separated values file will be loaded into the environment for preparation and subsequent analysis.

```
df <- read_csv('./zeek.csv', col_names = TRUE, show_col_types = FALSE)

dim(df)
```

```
## [1] 23748    98
```

The variables included in the raw data conform to GN unified data model standards and are far larger in number than this project requires. The publicly releasable data set consists of 23,747 observations across 98 features. This dataset can be subdivided to seven types (or classes) of logs: conn, dns, files, http, ssh, ssl, and weird (The Zeek Project, 2021). For each class of Zeek log, only a subset of the 98 features are applicable values. For this reason, whenever a particular log class is examined, the extraneous features should not be considered in order to avoid inserting artificial data sparsity (Shi et al., 2021).

```
df.conn <- df %>%
            filter(bdp.ingest.file.name == "conn.log") %>%
            select(all_of(conn.columns))

dim(df.conn)
```

```
## [1] 10727    34
```

The quality of the data is very high as the Zeek logs selected for study are intended to be used for forensic analysis applications by cybersecurity professionals. Overall sparsity within the selected log class (conn) is 12.4%.

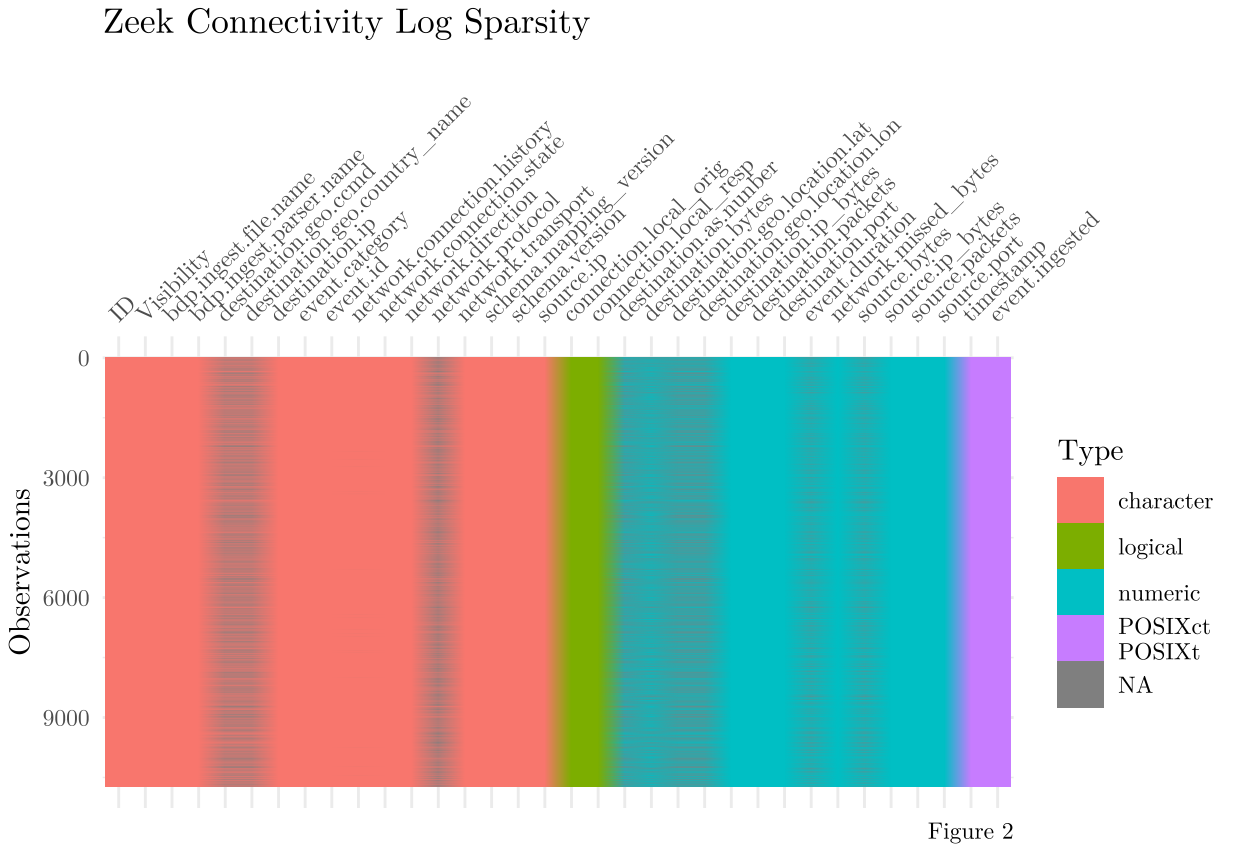

Figure 2

This agglomerative clustering project will focus on a select few features from the conn logs; specifically, this project will examine all 10,726 observations available in the conn logs across 4 specific features: destination.packets, destination.port, source.packets, and source.port.

```
df.conn.cut <- df.conn %>%
  select(destination.packets, destination.port, source.packets, source.port)

summary(df.conn.cut)
```

```
##  destination.packets destination.port source.packets      source.port
##  Min.   :   0.000    Min.   :   3.0   Min.   :    0.00   Min.   :    3
##  1st Qu.:   0.000    1st Qu.:  53.0   1st Qu.:    0.00   1st Qu.:40864
##  Median :   2.000    Median : 443.0   Median :    0.00   Median :47916
##  Mean   :   8.838    Mean   : 259.7   Mean   :   21.43   Mean   :47468
##  3rd Qu.:   5.000    3rd Qu.: 443.0   3rd Qu.:    0.00   3rd Qu.:54218
##  Max.   :1387.000    Max.   :5353.0   Max.   :48992.00   Max.   :60986
```

Data sparsity within the four selected features for analysis (destination.packets, destination.port, source.packets, and source.port) is 0%. Having a very low sparsity for the data is important as outlier detection could be skewed by missing values (Shi et al., 2021).

After reducing the original dataset down to the four chosen variables, a scaled version of the data will be created due to the differences in the ranges of the variables (Pamula et al., 2010).

```
df.conn.scaled <- as.data.frame(scale(df.conn.cut))
```

```
summary(df.conn.scaled)
```

```
##  destination.packets destination.port  source.packets      source.port
##  Min.   :-0.19245    Min.   :-1.0440   Min.   :-0.02502   Min.   :-5.7209
##  1st Qu.:-0.19245    1st Qu.:-0.8406   1st Qu.:-0.02502   1st Qu.:-0.7960
##  Median :-0.14890    Median : 0.7455   Median :-0.02502   Median : 0.0540
##  Mean   : 0.00000    Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.:-0.08357    3rd Qu.: 0.7455   3rd Qu.:-0.02502   3rd Qu.: 0.8136
##  Max.   :30.01203    Max.   :20.7143   Max.   :57.17936   Max.   : 1.6293
```

Outliers are frequently removed for traditional clustering analyses; however, they will not be removed for this project because the desired outcome in this case relies on the existence of correlations among the outliers that might identify threat actors' actions within the monitored network (Mazarbhuiya, 2019). As such, no further action will be taken to clean or prepare the data for analysis.

Part IV: Analysis

Distance Matrix

Having prepared the data for analysis, a distance (or dissimilarity) matrix will be calculated in order to compare four common linkage methods. The data was scaled to have a mean of 0 and a standard deviation of 1 during pre-processing, so we will use a Euclidean distance calculation.

```
df.conn.dist <- dist(df.conn.scaled, method = "euclidean")
```

Agglomerative Coefficients

The agglomerative coefficient (AC) of four common linkage methods—complete, average, single, and Ward's method—will be calculated and compared in order to determine which will be used for this project's analysis. The agglomerative coefficient conveys a sense of how defined the clustering structure is within a given data set. AC values closer to 1 suggest a more clearly defined structure, and less clearly defined structures exhibit lower AC values.

```
library(cluster)
set.seed(214)

hc.complete <- agnes(df.conn.dist, method = "complete")
hc.average  <- agnes(df.conn.dist, method = "average")
hc.single   <- agnes(df.conn.dist, method = "single")
hc.ward     <- agnes(df.conn.dist, method = "ward")
```

Average AC	Single AC	Complete AC	Ward's AC
0.9999928	0.9999898	0.9999935	0.9999964

We can see above that each of the calculated linkage methods produce values within .0001 of 1. This indicates a significantly well-defined clustering structure within the data regardless of the linkage method chosen. Ward's method will be employed for the remainder of this project due to being larger—even if only marginally—than its alternatives.

Number of Clusters

Dendrogram

Clustering analyses generally include a discussion of the dendrogram as a visualization that traces the progress of the clustering process. This visualization can—under ideal circumstances—help to discern the optimal number of clusters for a given data set. It can also increase confidence in the chosen linkage method where the dendrogram does not show anomalous behavior (e.g., dendrogram inversion). However, in this case, we see below that the significant class imbalance (many more non-outliers than outliers) makes determining the optimal number of clusters a somewhat trivial matter. By design, the desired outcome of this analysis is to distinguish between two classes of network traffic; hence, this result is expected and serves to reinforce the validity of the method and parameters chosen to support it (Zhang et al., 2019).

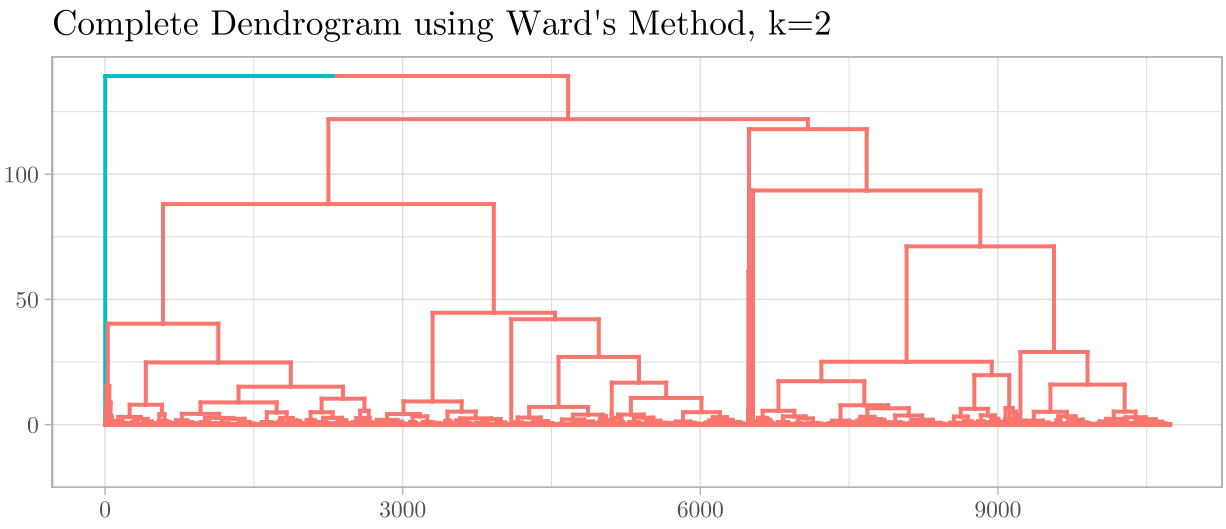

Figure 3

Elbow and Silhouette Plots

Another common method employed to validate the optimal number of clusters is to use one of several statistic graphs—among them, within-cluster sum of squares (WSS) and silhouette width statistic. Shown below are the so-called elbow (employing the WSS statistic) and average silhouette methods.

In using the elbow method, the WSS is calculated as a function of the number of clusters with an aim to minimize the total intra-cluster variation. The various WSS values can be compared to other WSS values produced by the same data set with a different k, but these statistics do not generalize well. For data sets that cluster such that WSS values function well, the optimal number of clusters will be shown by a bend in the plot that marks the appropriate number of clusters for that data set (Kaufman & Rousseeuw, 1990).

Alternatively, a silhouette width analysis assesses how well each individual observation fits in with its assigned cluster and estimates how far apart different clusters are. When these values are averaged and plotted as function of the number of clusters, the quality of the clusters—in terms of tightness and separation—can be assessed. Unlike the WSS statistic, the average silhouette width does generalize well. Values closer to 1 indicate very well defined and separated clusters while low (or negative) values indicate cluster overlap and likely erroneous membership assignment (Lengyel & Botta-Dukát, 2019).

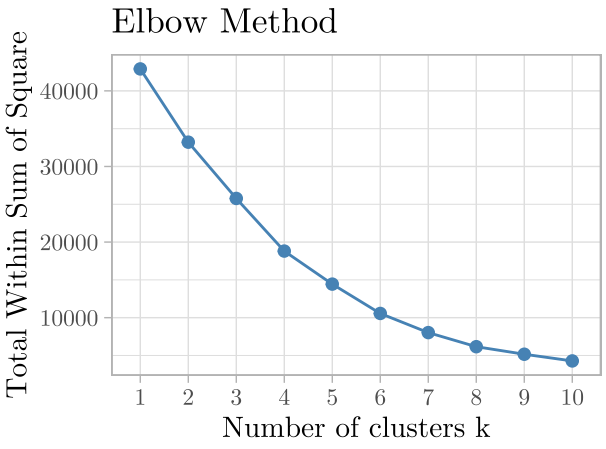

Figure 4

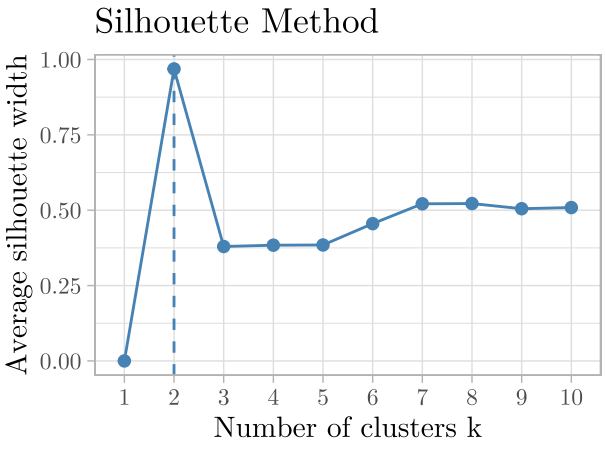

Figure 5

The elbow plot is largely inconclusive; there is no significant bend that might indicate an optimal cluster count. The silhouette statistic plot, however, is conclusive; it indicates that a two-cluster solution is optimal for this data set.

The project will proceed with two clusters. The choice—in this case—is supported by examining the silhouette statistic and dendrogram, which would be consistent with a typical approach for a normal hierarchical clustering analysis project. However, even if this approach were not conclusive, a two-cluster decision would still be the proper way forward for the intended application of this analysis. Network traffic is either normal or abnormal. While there may be several classes of normal (or abnormal) traffic, the desired outcome for the scope of this study that traffic that is the least similar to the majority of baseline traffic be flagged for human analysis.

Cluster Membership

Given the analytic objectives of this project and the average silhouette width graph, a two-cluster model will produce the optimal results. Below is a cluster plot showing the relative proximity of grouped network traffic. This visualization is effective at imparting a sense of the scale of the dissimilarity of the detected outliers.

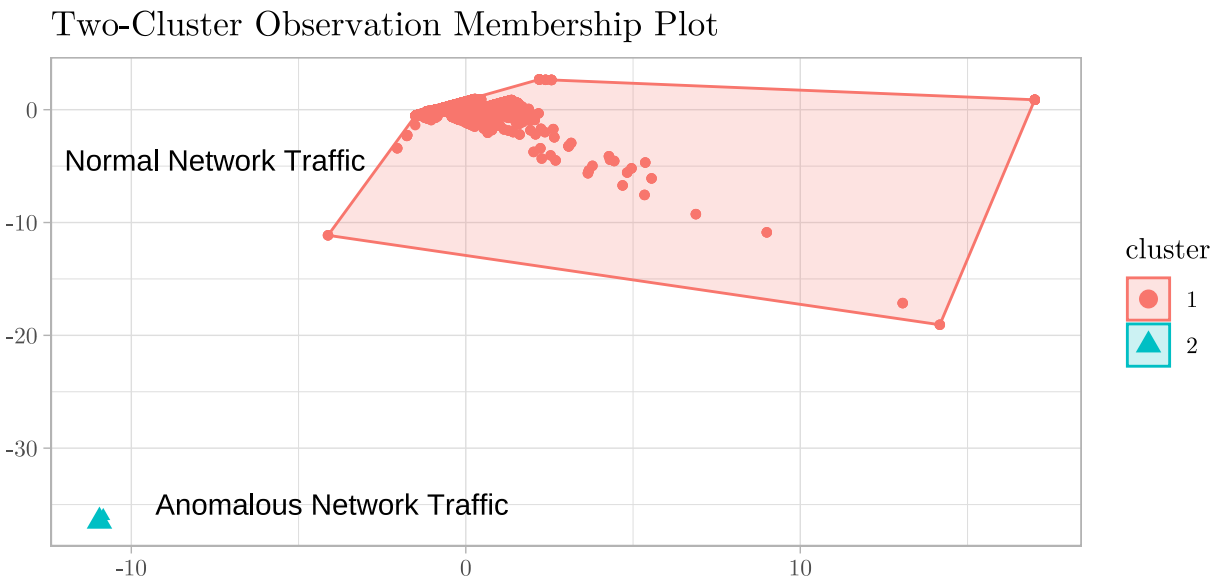

Figure 6

Figure 6, above, shows that there is one large cluster of traffic and one much smaller cluster of anomalous traffic. The observations classified by the model into cluster 1 will be inferred to be more likely normal, authorized, and typical than the observations classified by the model into cluster 2. We notice, again, that most of the network traffic is assigned to cluster 1 while a very small minority of observations fall into cluster 2. These results are expected and desired given the nature of the data and primary goal of identifying the network traffic observations that are the least similar to all other network traffic.

It would be useful to know precisely how many observations are anomalous (those associated with cluster 2) because those observations will be forwarded to cybersecurity analysts for further investigation as potential indicators of network compromise. The number of observations identified as such will determine the number of analysts needed, the amount of time required to investigate the nature of the suspect traffic, and the potential severity of the compromise.

```
table(clust.2)
```

```
## clust.2
##     1     2
## 10724     3
```

Here, the cluster containing the anomalies has a cardinality of 3. Subjectively, these 3 observations are well-separated and tightly grouped. This lends confidence to the inference that each of these observations are similar in nature and likely related. However, as objective measurement is available to assess the quality of the groupings.

Silhouette Values

The silhouette width values will be produced and shown here to assess the objective quality of the two clusters. This method is advantageous "when one is seeking compact and clearly separated clusters" (Rousseeuw, 1987). Given the scope of this project, such a separation of clusters is best suited for anomaly detection—making this method of selection of the optimal number of clusters for this project ideal. This method computes the silhouette coefficients for each individual point to determine the cohesion (how well that single object fits in with its assigned group) and separation (how well distanced that single object is from the other groups).

The range of the silhouette values is $[1, -1]$. A value closer to 1 indicates that the object fits in well within its assigned cluster and is a poor match to alternative clusters. Low or negative values indicate that an observation is assigned to a cluster in which it does not fit well. A silhouette plot will rank the silhouette values of every observation from highest to lowest (left to right), and the curve and fall-off on the right side of each cluster will indicate the ratio of its observations that are well vice poorly fit.

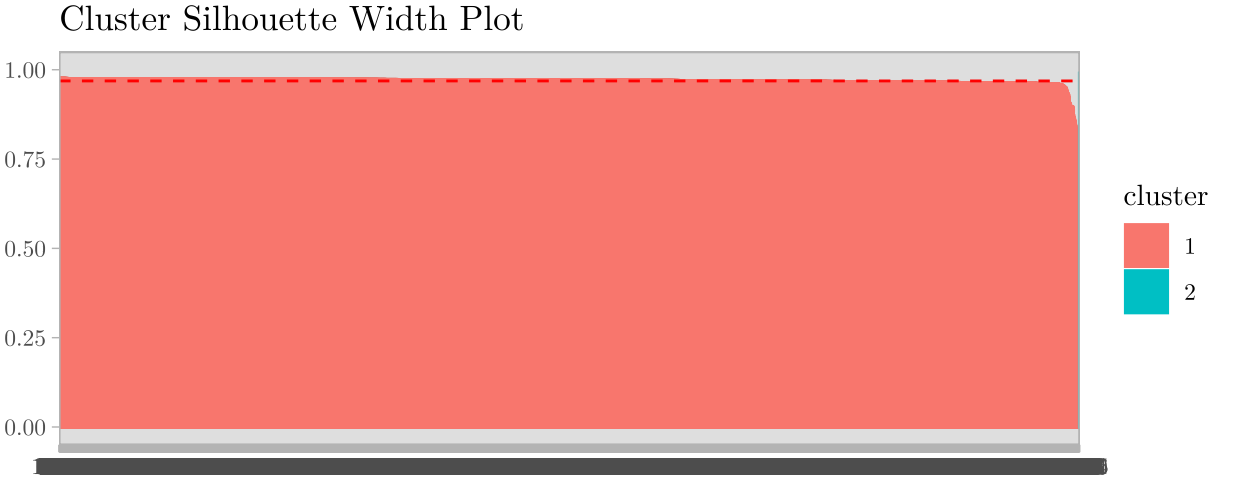

Figure 7

Figure 7, above, shows cluster 1 to be a solid and nearly rectangular block with very little falloff on the right side. The interpretation here, particularly with an average width of 0.9686805 is that the two clusters produced by the methods selected above have high degrees of cohesion and separation. The 3 observations from cluster 2 are not easily visible in the plot; however, the cluster 2 average silhouette width is 0.9870862—even more tightly defined than cluster 1.

Technique Assessment

With any machine learning approach, there are going to be strengths and weaknesses associated with the decisions necessary to achieve the desired result. It is worthwhile to address advantages and disadvantages of these techniques in order to present a complete picture to those that might implement or iterate on the methodologies concerned.

Advantage: Network Defined Normal

Due to how an agglomerative hierarchical clustering algorithm functions, each application of this method will use the traffic patterns of the analyzed network to establish appropriate thresholds for outlier definitions. This means that the approach could be applied with very similar workflows across various industries and network topographies. Much like the appeal of anomaly detection vice signature matching, this unsupervised data mining approach to forensic network analysis generalizes very well (Soliman et al., 2021).

Disadvantage: Computational Cost

For this project, a Zeek log pull for a single ninety-minute span was examined. Despite this narrow scope, the virtual machine running a production instantiation of RStudio Server had to be shut down and upgraded in order to maintain stability during the data processing steps required to complete the analysis identified by the research question in Part I. Even after the upgrade (which resulted in additional costs per running hour), secondary projects had to be offloaded to alternate servers. The computational cost considerations of this procedure are not negligible.

It is not necessarily the case that a network administrator will be able to narrow down a set of logs requiring analysis to such a narrow window in all circumstances. While some threats sent atypical traffic across a network with a high degree of consistency, some advanced persistent threats (APTs) are designed to minimize their signatures to reduce the detection surface presented to anomaly detection techniques (Ghafir & Přenosil, 2014). Despite being functioning robustly in an environment affected by a high class imbalance, a business employing this technique must weigh the costs of consuming IT assets, cloud resources, and time against the benefits of identifying potential compromises (Zhang et al., 2019).

Part V: Summary and Implications

Analysis Results

The goal of this project was to separate the Zeek log data into multiple clusters that group legitimate network traffic types together and isolate anomalous traffic that might indicate a compromise of the network. The agglomerative coefficients produced by multiple linkage methods were greater than 0.9 and the average silhouette width for our final clustering model was also greater than 0.9. The smallest cluster contained three observations (0.02% of total observations) and a silhouette width of greater than 0.99. These results necessitate a rejection of the null hypothesis that an agglomerative hierarchical clustering analysis cannot reveal anomalous network behavior. Similarly, this study's results necessitate an affirmative response to the research question of whether an agglomerative hierarchical clustering analysis of network Zeek logs can function as an effective anomaly detection technique.

Analysis Limitation

Anomaly detection—while a valuable part of the threat identification process—is not equivalent to threat identification on its own. It is true that legitimate network traffic can be anomalous. For example, an authorized systems administrator installs an approved new program that establishes verification and activation telemetry with a licensing server. That kind of traffic will likely use atypical ports and contain unusual packet bursts that would be flagged (correctly) as anomalous; however, it would not be an indicator of compromise.

Within the defined scope of this project, the anomalies are not necessarily assessed for maliciousness. For this reason, the structure of the analysis calls for the forwarding of flagged traffic to cybersecurity specialists for further analysis.

Course of Action Recommendation

In a circumstance where it is not possible or otherwise impractical to leverage the real-time scanning of a NIDS, a network administrator may still want to assess the security state of a network with available log data. Rather than be required to sift through the entirety of large amounts of traffic or apply domain knowledge to a very large dataset, it may be desired to narrow the scope of data that a human specialist should examine such that the bulk of benign traffic does not have to be considered.

For this scenario, it is recommended that an agglomerative hierarchical clustering technique be applied to the overall count and port information from both the source and destination of packets from the network's connectivity Zeek logs. This study found that using Ward's method criterion was the most effective linkage method. A silhouette statistic graph can be used to determine the theoretically optimal number of clusters; however, a two-cluster solution will best separate the typical network traffic from the traffic that least conforms to the norms of the tested environment. Upon completion of the agglomerative clustering analysis, the traffic flagged as outliers should be forwarded to the relevant cybersecurity specialists for manual review.

Approaches for Further Study

The results of this study indicate a successful proof-of-concept for a specific machine learning enabled security analysis method where real-time network monitoring is not possible. However, the method demonstrated here can likely be improved or otherwise optimized for simpler or more reliable implementation.

Feature Combinations

Despite experimenting with a multitude of variable combinations while conducting this study, the set of tested feature combinations is far from exhaustive. This scope of the findings of this project do not claim certainty that an alternative feature selection would not produce more significant or more directly useful clusters. Additional research should be conducted with varying features from the connectivity logs or from other Zeek log classes. Particular attention should be focused on increasing the separation of the outliers from legitimate network traffic while keeping the authorized traffic from being flagged as a false positive.

Threat Ranking

This approach requires a cybersecurity analyst to examine each of the outliers identified as potential evidence of the presence of a threat actor within their network. A further avenue of study could be to run a similar agglomerative hierarchical clustering algorithm on a larger number of other data sets, group all the identified outliers, label the data as benign or malicious, and then attempt to train a secondary algorithm to prioritize the flagged outliers according to their relative likelihood of being an actual indicator of network compromise.

References

Andrews, D., Behn, J., Jaksha, D., Seo, J., Schneider, M., Yoon, J., Matthews, S., Agrawal, R. & Mentis, A. (2019). Exploring RNNs for analyzing zeek HTTP data. HotSoS '19: Proceedings of the 6th Annual Symposium on Hot Topics in the Science of Security, 1–2. https://doi.org/10.1145/3314058.3317291

Brust, A. (2021). Army hopes Big Data Platform enables deeper analysis across bigger datasets. Federal News Network. Retrieved July 12, 2022, from https://federalnewsnetwork.com/technology-main/2021/08/army-hopes-big-data-platform-enables-deeper-analysis-across-bigger-datasets/

Ghafir, I., Hammoudeh, M., Přenosil, V., Han, L., Hegarty, R., Rabie, K.M., & Aparicio-Navarro, F.J. (2018). Detection of advanced persistent threat using machine-learning correlation analysis. Future Gener. Comput. Syst., 89, 349-359. https://doi.org/10.1016/j.future.2018.06.055

Kaufman, L. and Rousseeuw, P.J. (1990) Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York. http://dx.doi.org/10.1002/9780470316801

Keells, J. (2021, July 9). Python vs R for data science. Medium. Retrieved July 12, 2022, from https://medium.com/octave-john-keells-group/python-vs-r-for-data-science-320e167ffe90

Lengyel, A., & Botta-Dukát, Z. (2019). Silhouette width using generalized mean—A flexible method for assessing clustering efficiency. Ecology and Evolution, 9(23), 13231–13243. https://doi.org/10.1002/ece3.5774

Mazarbhuiya, F.A., AlZahrani, M.Y., Georgieva, L. (2019). Anomaly Detection Using Agglomerative Hierarchical Clustering Algorithm. In: Kim, K., Baek, N. (eds) Information Science and Applications 2018. ICISA 2018. Lecture Notes in Electrical Engineering, vol 514. Springer, Singapore. https://doi.org/10.1007/978-981-13-1056-0_48

Newton, H., & Schoen, S. (2022). Newton's Telecom Dictionary (32nd ed.). Telecom Publishing.

Pamula, R., Deka J. K., Nandi, S. (2011). An Outlier Detection Method Based on Clustering, Second International Conference on Emerging Applications of Information Technology, 2011, pp. 253-256, doi: 10.1109/EAIT.2011.25. https://doi.org/10.1109/EAIT.2011.25

Rousseeuw, P. (1987). "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis." Journal of Computational and Applied Mathematics 20. Elsevier: 53–65. https://doi.org/10.1016/0377-0427(87)90125-7 Shi, P., Zhao, Z., Zhong, H., Shen, H., & Ding, L. (2021). An improved agglomerative hierarchical clustering anomaly detection method for scientific data. Concurrency and Computation: Practice and Experience, 33. https://doi.org/10.1002/cpe.6077

Sikos, L. (2020). Packet analysis for network forensics: A comprehensive survey. Forensic Science International: Digital Investigation. Volume 32. https://doi.org/10.1016/j.fsidi.2019.200892

Soliman, H. M., Salmon, G., Sovilj, D. & Rao, M. (2021). RANK: AI-assisted end-to-end architecture for detecting persistent attacks in enterprise networks. CoRR, abs/2101.02573. https://doi.org/10.48550/arXiv.2101.02573

The Zeek Project. (2020). The Zeek Network Security Monitor. https://zeek.org/

The Zeek Project. (2021). conn.log — Book of Zeek (git/master). Zeek Logs Official Documentation. Retrieved July 12, 2022, from https://docs.zeek.org/en/master/logs/conn.html

Wickham, H., & Grolemund, G. (2017). R for Data Science. Sebastopol, CA: O'Reilly. https://r4ds.had.co.nz/

Zhang, Y., Liu, J., Li, H. (2010). An Outlier Detection Algorithm Based on Clustering Analysis. First International Conference on Pervasive Computing, Signal Processing and Applications, 2010, pp. 1126-1128, doi: 10.1109/PCSPA.2010.277. https://doi.org/10.1109/PCSPA.2010.277

Zhang, Y., Liu, J., Zheng, L., & Yan, C. (2019) A Hierarchical Clustering Strategy of Processing Class Imbalance and Its Application in Fraud Detection. IEEE 21st International Conference on High Performance Computing and Communications. IEEE 17th International Conference on Smart City; IEEE 5th International

Conference on Data Science and Systems (HPCC/SmartCity/DSS). https://doi.org/10.1109/HPCC/SmartCity/DSS.2019.00249