Clustering of Multistage Cyber Attacks using Significant Services

Christopher T. Murphy
Department of Computer Engineering
Rochester Institute of Technology
Rochester, NY, USA
ctm3494@gmail.com

Shanchieh Jay Yang
Department of Computer Engineering
Rochester Institute of Technology
Rochester, NY, USA
jay.yang@rit.edu

Abstract – *Multistage cyber attacks may target services of different types, often indicating their behavior or capability of penetrating into the network. A significant enhancement to network defenses will be to recognize the different classes of multistage attacks, allowing timely and effective anticipation of future attacks. Drawing analogies from social networking analysis, this work proposes a methodology that clusters cyber attacks based on the 'significant services' being exploited. From transforming the attacked services to utilizing the Divisive Hierarchical Clustering algorithm, the proposed method is able to identify sub-communities of attacks that share common characteristics. Experiment results demonstrate a high modularity for the identified community structure. Novel discoveries are also made possible by examining the attack clusters and the resulting dendrogram.*

Keywords: Cyber Attacks, Clustering, Social Network, Modularity.

1 Introduction

Network analysts are bombarded with large amounts of low level data, posing great challenges for them to differentiate and recognize critical multistage cyber attacks. Multistage attacks are performed by hackers to compromise one or more machines in a network to gradually gain access to critical information or network operation hidden behind layers of firewall rules. These attacks, composed of correlated Intrusion Detection System (IDS) alerts, can be diverse in their capabilities to penetrate the network. There exists no current literature defining how to classify or categorize these diverse attacks. By fusing information on a higher level, this work aims to perform unsupervised learning to cluster and identify types of multistage attacks.

An attack may exploit and compromise different types of network services, often indicating their capability or preference in penetrating the network. By treating the attacked services as the features of cyber attacks, this work develops a system that effectively uncovers characteristics that define similar attacks. By drawing analogy to social network analysis, Divisive Hierarchical Clustering is utilized to cluster attacks, forming optimal community structure with respect to the similarity in attacked services. By identifying clusters of attacks, the system provides a means to comprehend and differentiate the potentially many concurrent malicious activities. Analysts or attack projection algorithms, e.g., [1, 2], can analyze attacks that exhibit similar capability or preference, and provide a more accurate assessment.

The rest of the paper is organized as follows. Section 2 briefly discusses the various clustering algorithms and their suitability for cyber attacks. Section 3 presents the technical challenges and the proposed approach. A set of attack scenarios were tested to demonstrate the clustered attacks, and Section 4 discusses these experiments and the observations, followed by the conclusion in Section 5.

2 Related Work

To our knowledge, there is no existing work explicitly classifying or clustering multistage cyber attacks. Note that this is different from the low level problem where attacks are identified and categorized into, e.g., Denial-of-Service attacks or buffer overflow attacks. This work tackles the problem of characterizing and clustering the cyber attacks based on the exploits each attack has executed on potentially many machines on a network. A brief summary of clustering techniques used for different applications are presented next.

There are two main classes of clustering algorithms, partitional and hierarchical [3]. Partitional clustering, e.g., k-means clustering, requires a predefined number of clusters as an input, which may not be feasible for any given network. In fact, our results exhibit that the number of clusters varies significantly from one scenario to another. There are two main types of hierarchical clustering [4]: divisive hierarchical clustering and agglomerative hierarchical clustering. Divisive hierarchical clustering has been argued to be superior over agglomerative hierarchical clustering because (1) agglomerative hierarchical clustering sometimes fails to find known sub-communities, and (2) "core nodes in a community are connected early in the agglomerative process, but peripheral nodes tend to get neglected" [4].

In its essence, Divisive Hierarchical Clustering (DHC)

iteratively remove edges from a graph, $G(V, E)$, to find the optimal sets of connected vertices, with each set representing a sub-community. DHC has been shown to successfully find sub-communities in professional collaboration networks [4], bottlenose dolphins in Doubtful Sound [4], NCAA football teams [5], karate club members [4], characters in *Les Miserables* [4], books on American politics [6], and food webs [5]. In many of these cases, the sub-communities found can be verified against common knowledge. For example, sub-communities were found for the professional collaboration networks based on both geographic location and intellectual descendancy, and college football teams were clustered mostly into their respective conferences. An important step for successfully finding sub-communities using DHC is to determine a good initial graph, particularly in terms of defining the edges. In the case of clustering cyber attacks, it is unclear what the initial graph should entail and the nature of sub-communities one expects to find. The following section addresses these questions.

3 Methodology

A multistage cyber attack consists of one or more observations, e.g., IDS alerts. These alerts typically contain attributes such as source IP, target IP, and a description of the exploit used. The exploit description can be associated with a service by referring to online databases such as the National Vulnerabilities Database (NVD) [7] or the Common Vulnerabilities and Exposures (CVE) Dictionary [8]. Given a set of attacks, the goal is to determine a graph $G(V, E)$ where each edge $e \in E$ represents some similarity between a pair of attacks $v \in V$. This similarity will be defined in respect to attacked services in this work. Other alert attributes are possible and will be discussed in Section 5. Once an initial graph is defined, DHC can be applied to find the sub-communities within the initial graph G.

3.1 Similarities based on Attacked Services

An attack service matrix (ASM), $A_{m \times n}$, is defined to represent the number of times each of the m attacks has executed exploits toward one of the n services. An example attack service matrix is shown in Table 1. For example, attack track 8 has attacked the FTP service and the HTTP service twice each, NetBIOS server once, and Telnet once. Note that each attack may have a different number of observables, and some services may be more commonly and repeatedly attacked than others. The ASM provides raw data about the capability and behavior of the attacks but it is unclear which attacks are similar. More specifically, it is unclear which elements in the ASM are 'significant' enough to differentiate between the attacks.

Collaborative Filtering [9] has been used to find similarities between rows of values within a matrix. For example, it has been used to find similar movie preferences based on user rankings of movies [10]. The Pearson correlation coefficient [9] $(w_{i,j})$ is commonly used to represent the similarity between rows i and j of a given matrix R. Let $r_{i,k}$ be the (i, k)th element of R and $\bar{r}_i$ is the average among all elements on row i.

$$w_{i,j} = \frac{\sum_k (r_{i,k} - \bar{r}_i)(r_{j,k} - \bar{r}_j)}{\sqrt{\sum_k (r_{i,k} - \bar{r}_i)^2 \sum (r_{j,k} - \bar{r}_j)^2}} \quad (1)$$

A drawback of using the Pearson correlation coefficient is that it assumes that $r_{i,k}$ falls within a given strict scale (e.g., movie ranking from 1 to 5), and works well when the elements in the matrix R exhibit a bell curve distribution - a common result of finite scales. In an ASM, an attack can have orders of magnitude more observables than other attacks, and a service can be attacked many times more than another service. The distribution of the elements from the ASM example shown in Table 1 is given in Figure 1. It is clear that there is a heavy bias of services being attacked few times by any attack - not much of a bell curve. In order to overcome this problem, a transformation of the ASM is needed.

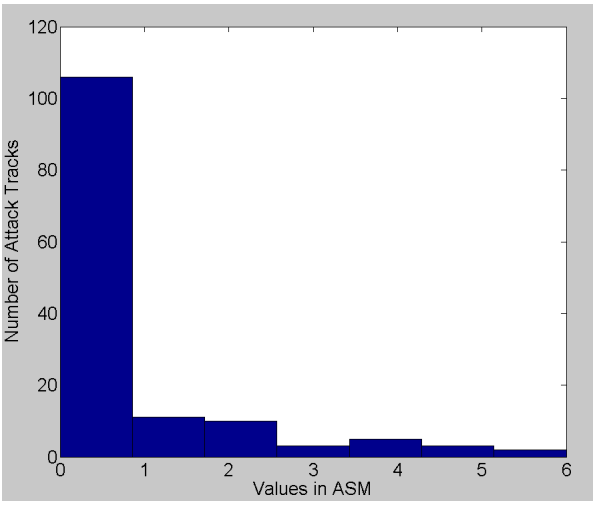

Figure 1: Distribution of the values in the example ASM shown in Table 1.

Anscombe and Tukey [11] proposed an elegant method to differentiate significant elements and outliers in a matrix. Specifically, a residual value, $R_{i,j}$, is the difference between the observed value, $A_{i,j}$, and the *fitted value*, $F_{i,j}$,

$$R_{ij} = A_{ij} - F_{ij} \quad (2)$$

where

$$F_{ij} = \bar{A}_{i.} + \bar{A}_{.j} - \bar{A} \quad (3)$$

and $\bar{A}_{i.}$ is the mean of the ith row, $\bar{A}_{.j}$ is the average of the jth column, and $\bar{A}$ is the average of the entire ASM. A positive residual value in the matrix is indicative of a higher than expected value in the entire ASM, and a negative value is indicative of a lower than expected value. The distribution of the values in the residual matrix created from the example

	FTP Server	SMTP Server	POP3 Server	HTTP Server	Service A	Service B	NetBIOS Server	RPC Server	IMAP Server	DNS Server	Telnet	SSH	AIM	SQL
Attack 1	4	4	2	2	0	0	0	0	0	0	0	0	0	0
Attack 2	0	2	0	3	0	0	0	0	0	0	0	0	0	0
Attack 3	5	0	0	0	0	0	0	0	0	0	0	0	0	0
Attack 4	0	1	0	2	0	0	0	0	0	0	0	0	0	0
Attack 5	5	0	0	1	0	0	0	0	0	0	0	0	0	0
Attack 6	4	1	0	4	0	0	2	2	0	0	1	0	0	0
Attack 7	2	0	0	1	0	0	0	0	0	0	0	0	0	0
Attack 8	2	0	0	2	0	0	1	0	0	0	1	0	0	0
Attack 9	6	4	1	3	1	0	0	0	0	0	5	0	0	0
Attack 10	2	1	0	6	1	0	0	0	0	0	1	0	0	0

Table 1: An example *Attack Service Matrix (ASM)*.

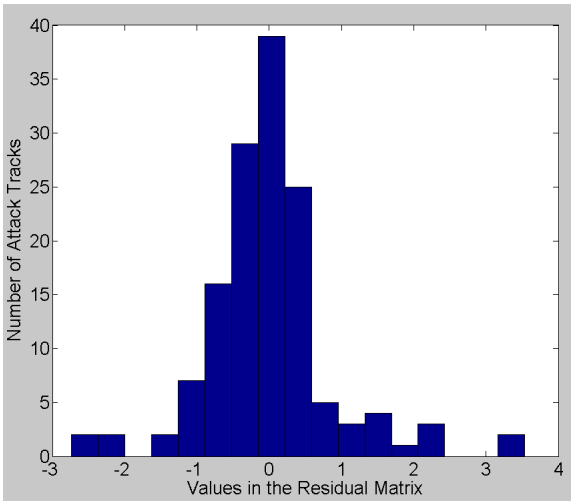

Figure 2: Distribution of the residual values created from the example ASM shown in Table 1.

ASM is shown in Figure 2, which exhibits roughly a bell curve.

The residual matrix R is then applied to (1) and determines the similarity matrix W, which is a symmetric n by n matrix measuring the closeness of each pair of attacks in terms of the attacked significant services. From the aforementioned example ASM, the similarity value between Attacks 7 and 8 is 0.69 - a high number due to both attacks exploiting the FTP and the HTTP Services. Even though neither attacks attack either service many times, both services were significant to the two attacks. On the contrary, Attacks 1 and 8 have almost nothing in common, and consequently, the similarity is -0.80.

3.2 Social Network Graph of Cyber Attacks

Directly using the similarity matrix to define a social network graph of cyber attacks gives a weighted graph, which significantly increases the computation complexity for clustering. A thresholding approach is adopted to transform W into a 0-1 adjacency matrix G where

$$C_{i,j} = \begin{cases} 1 & \text{if } W_{i,j} \geq \beta \\ 0 & \text{if } W_{i,j} < \beta \end{cases} \quad (4)$$

The choice of β should satisfy two desired characteristics for the social graph: (1) the number of connected vertices is maximized, and (2) the number of edges is minimized. This is because that DHC was designed for sparse yet connected networks [5].

It has been shown that a planar graph with a minimum degree of 3 has a high probability of being connected [12]. Therefore, an analysis of the number of 3-connected vertices as β increases can determine the choice of β. Considering one of the example data sets (to be illustrate in Section 4), Figure 3 shows a sharp drop of the number of 3-connected vertices as β increases to around 0.6. Consider a straight line segment that begins and ends at the same points as the declining curve. The value of β (0.67 in this case) is chosen at the point where it exerts the greatest difference between the curve and the straight line segment.

3.3 Divisive Hierarchical Clustering

The adjacency matrix C defines the initial graph for DHC to perform clustering. DHC iteratively removing edges that lead to partitions where the vertices (i.e., the attacks) reside in sub-communities. A vertex is in the same sub-community of another vertex if and only if there exist a path between the two vertices. A *partition* is the result of each edge removal, and a metric called *modularity* (Q) [13] can be used to measure the quality of the community structure for each

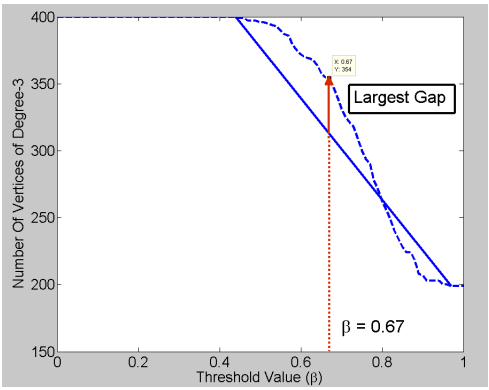

Figure 3: The number of vertices of degree-3 in attack social graph as β increases.

partition.

$$Q = \sum_{s=1}^{p} \left[\frac{l_s}{L} - \frac{d_s}{2L}^2 \right] \quad (5)$$

where p is the number of sub-communities in a particular partition, l_s is the number of edges within the sub-community s, L is the total number of edges in the initial graph, and d_s is the degree of the nodes in sub-community. Calculating the modularity for all possible partitions of a graph is believed to be an NP-complete problem [4]. Consequently, DHC was designed to heuristically determine which edges to remove and in what order to remove, in order to find a good partition in each iteration.

To choose which edges to remove, the *edge betweenness* is calculated for every edge. The edge betweenness algorithm works by finding the shortest path between each pair of vertices in the graph. Every time an edge is found along a shortest path, its edge betweenness is incremented by one. Newman and Girvan [4] pointed out that the sub-communities are supposedly connected by very few edges and therefore paths between vertices will be along one of these edges [4]. Naturally, the paths between the communities will have the largest edge betweenness, and therefore, are the edges to remove. The iterative process gives finite number of (modular) partitions to consider and the one with the highest Q will be chosen to reveal the sub-communities or clusters of cyber attacks.

4 Experiments and Results

4.1 Experiment Setup

In order to test our methodology, a test network was created using the simulator developed by Costantini [14]. The simulator allows for variations in efficiency, noise, target IP, and other simulation parameters. The test network was designed to have hierarchical levels modeled after an academic institutes's organization. The network is divided into two main subnets: (1) Financial and (2) Academic, with the academic subnet further broken down by colleges (Art, Science, Engineering). The engineering college is, in turn, further broken down into Computer, Electrical, and Mechanical Engineering Departments. Throughout the network, there are multiple labs and servers running a variety of services appropriate for the location in the network. For example, Oracle Web Application Servers is only found on the Financial subnet. Also, there are more servers in the Engineering College (especially Computer Engineering) and are consequently more specialized than the Art College. The network details are further explained in Table 2.

Two data sets were created and used for testing our methodology. The first data set, Data Set 1, contains 400 cyber attacks representing *efficient* and *inefficient* attack scenarios, all with specifically selected targets. The purpose of this data set is to show how the DHC is able to distinguish between efficient and inefficient attacks, and make observations from the resulting clusters. A summary of the attacks in Data Set 1 are shown in Table 3. Data Set 2 contains attacks that are more 'randomly' specified with targets over the entire network along with other attack parameters. It was designed to demonstrate how our method can discover the different types of cyber attacks that can happen in the network. The attacks in Data Set 2 are summarized in Table 4.

Target	Efficient Attacks	Inefficient Attacks
Math Lab	50	50
Physics Lab	50	50
Organic Lab	50	50
Inorganic Lab	50	0
CEDA Lab	0	50

Table 3: Data Set 1 Attack Parameters.

Target	Efficient Attacks	Inefficient Attacks
Math Lab	5	5
Physics Lab	5	5
Organic Lab	5	5
Inorganic Lab	5	0
CEDA Lab	10	10
RITZ	5	0
Flex Server	0	5
Debit Server	5	0
Art Server	5	0
VLSI Server	0	5
CE File Server	5	0
Engineering Email Server	0	5
Registration Server	5	0
External DNS Server	5	0

Table 4: Data Set 2 Attack Parameters.

hbt

Department	Number of Servers	Number of Host Clusters	Number of Services
External	4	0	20
Academic	2	0	6
Art	1	1	19
Engineering	1	0	5
Computer Engineering	3	2	9
Mechanical Engineering	1	1	10
Electrical Engineering	1	2	11
Science	5	4	5
Financial	5	4	16

Table 2: Table Describing Network Setup.

4.2 Results

Analysis presented here focuses on (1) the dendrogram, (2) the resulting modularity, and (3) the contents of the resulting clusters. For Data Set 1, the resulting cluster contents are shown in Table 5. There were a total of 32 clusters with a modularity of 0.701. In [4], the authors suggested that a modularity between 0.3 and 0.7 is typical for social networks. With a modularity value of 0.701, we can be confident that the clusters produced show strong modularity and community structure.

Twenty of these 32 clusters contain a total of 22 inefficient attack tracks. Four other clusters contained 135 inefficient attacks and no efficient attacks. These 24 clusters contained a total of 157 inefficient attacks and no efficient attacks, which means over 75% of the randomly generated inefficient attacks were clustered separately from the efficient attacks. The remaining eight clusters contained both efficient and inefficient attacks, but upon closer inspection, the suposedly inefficient attacks in these eight clusters actually showed efficient properties. The random generation of attacks are bound to create outliers with specified parameters. Table 5 shows the common sequences of attacked services for these clusters.

For Data Set 2, there was a total of eight clusters produced with a modularity of 0.702, which again implies a good community structure. The sole attack track in Cluster 1 and Cluster 2 were each unique and consequently separated from other attacks. Table 6 shows the contents of the clusters, several interesting observations were made. In particular, cluster 6 contained a large number of attacks on SMTP, IMAP, and POP3, the three email protocols. Although the proposed framework knows nothing about how these three service are similar, DHC was able to cluster them together because of the similar behaviors. The Oracle Web Server Application was only found on the Financial subnet of the network, and almost every attack that attacked this service was clustered together in Cluster 3.

In the process of DHC, a 'dendrogram' is produced to show a more detailed hierarchical structure between the attacks. While the sub-communities identified with the optimal modularity gives the clusters, the dendrogram shows the order over which the attacks are 'separated' from one another. Figure 4 shows the dendrogram associated with Data Set 2. Notice the 'staircases' appearing in the dendraogram. Each staircase represent a cluster and the stairs represent the order over which attacks are pulled off from the clusters they belong to. The dash lines shown in Figure 4 represent the separations between the clusters, i.e., where a critical edge is removed from the graph resulting in separation of sub-communities. Note that Clusters 1 and 2 (on the far right of Figure 4) are unique attacks and they actually have a degree of 0 in the initial graph. This can also be seen using the dendrogram as they are the first ones created during the iterative process of DHC. The appearance of the dendrogram can also be indicative of how certain clusters are related. For instance, Clusters 7 and 8 (the second and the third regions separated by the dash lines in) are similar in that both have attacks on the Apache Service (see Table 6), and in the dendrogram, they are next to each other and appears to belong to a common staircase structure.

5 Conclusion

The methodology presented in this paper aims at extracting information from multistage cyber attacks in order to characterize and cluster the attacks using Divisive Hierarchical Clustering. This is accomplished by creating an Attack Service Matrix, which quantifies the number of times each attack attacked each service. Residues are used to identify significant services in each attack, and the Pearson correlation coefficient is used to determine similarity between pairs of attacks. A threshold is applied to the similarity matrix to generate a sparse and mostly connected social network graph, making Divisive Hierarchical Clustering effective. Our results shows that the cyber attacks exhibit high modularity in terms of significant attacked services, and the clusters identified exhibit common attack characteristics. By identifying clusters of attacks with similar capabilities and preferences, one may anticipate a more efficient and accurate cyber impact and threat assessment.

Cluster	Attack Tracks	Feature
Cluster 1	12	RPC > TFTP > MySQL > MySQL > Apache > Apache > Windows
Cluster 2	11	RPC > TFTP > Windows
Cluster 3	12	RPC > FTP > MySQL > Apache > XWindows > Windows
Cluster 4 - 23	23	Twenty-two attack tracks in twenty clusters
Cluster 24	42	Forty efficient attacks and two inefficient attacks
Cluster 25	76	Almost all start with two attacks on the DNS service
Cluster 26	25	Apache > ColdFusion
Cluster 27	68	Start with FTP > RPC and large number of attacks on the Oracle Web Application Service
Cluster 28	21	SMTP and IMAP, which are both email protocol attacks
Cluster 29	15	SMTP > IMAP > Apache > MySQL > Apache > Apache > XWindows > XWindows > Windows
Cluster 30	42	All were inefficient with a high number of attacks on the Windows service
Cluster 31	4	noisy attack tracks that start with RPC > FTP and have also attacked the MySQL and Apache services numerous times
Cluster 32	50	Forty-four are efficient attacks, start with the service sequence of RPC > FTP > Apache > Apache

Table 5: Results from Data Set 1.

Cluster	Attack Tracks	Feature
Cluster 1	1	Only one attack track
Cluster 2	1	Only one attack track
Cluster 3	21	Began with attacks on RPC and FTP, contained almost every attack on Oracle
Cluster 4	25	Large number of DNS attacks
Cluster 5	5	Nothing remarkable about this cluster
Cluster 6	20	Large number of attacks on SMTP, IMAP, and POP3
Cluster 7	23	High number of Apache attacks
Cluster 8	4	RPC > FTP > Apache > XFS/ColdFusion

Table 6: Results from Data Set 2.

References

[1] J. Holsopple and S. Yang, "FuSIA: Future situation and impact awareness," in *Proceedings of 11th International Conference on Information Fusion*, 2008, pp. 1–8.

[2] D. Fava, S. Byers, and S. J. Yang, "Projecting cyber-attacks through variable-length markov models," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 359 – 369, 2008.

[3] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys (CSUR)*, vol. 31, no. 3, pp. 264 – 323, 1999.

[4] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, p. 026113, 2004.

[5] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *in Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[6] M. E. J. Newman, "Modularity and community structure in networks," *in Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[7] N. I. of Standards and C. S. D. Technology, "National vulnerabilities database (NVD)." [Online]. Available: http://nvd.nist.gov/nvd.cfm

[8] Mitre, "Common Vulnerabilities and Exposures CVE dictionary." [Online]. Available: http://cve.mitre.org/

[9] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: an open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 1994, pp. 175 – 186.

[10] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the 4th Conference on Uncertainty in Artificial Intelligence*, July 1998, pp. 43–52.

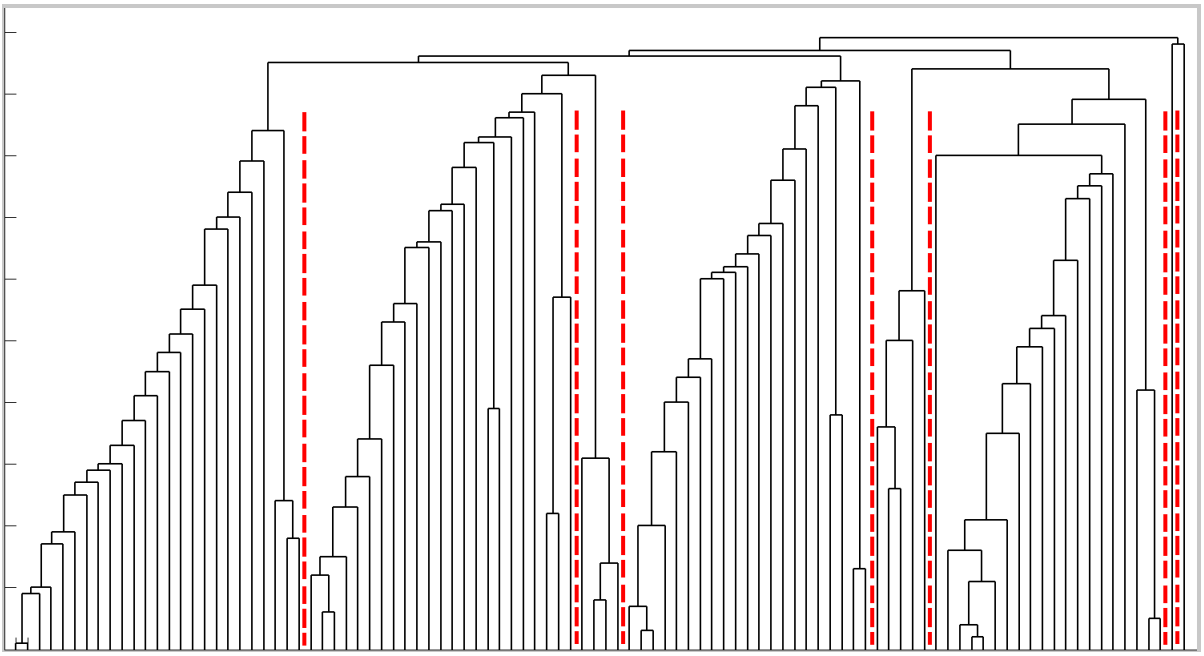

Figure 4: The resulting dendrogram produced from Data Set 2. The clusters are separated by the dash lines. From left to right are Clusters 4, 7, 8, 3, 5, 6, 1, and 2.

[11] F. J. Anscombe and J. W. Tukey, "The examination and analysis of residuals," *Technometrics*, vol. 5, no. 2, pp. 141 – 160, May 1963.

[12] M. Bodirsky, M. Kang, M. Löffler, and C. McDiarmid, "Random cubic planar graphs," *Random Struct. Algorithms*, vol. 30, no. 1-2, pp. 78–94, 2007.

[13] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *in Proceedings of the National Academy of Sciences*, vol. 104, no. 1, pp. 36–41, 2007.

[14] K. C. Costantini, "Development of a cyber attack simulator for network modeling and cyber security analysis," Master's thesis, Rochester Institute of Technology, 2007.