CHArcerization of Relevant Attributes using Cyber Trajectory Similarities

Jordan Bean

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation
CHAracterization of Relevant Attributes using Cyber Trajectory Similarities

by

Jordan Bean

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

Supervised by

Professor Dr. Shanchieh Jay Yang
Department of Computer Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York
November 2009

Approved by:

Dr. Shanchieh Jay Yang, Professor
Thesis Advisor, Department of Computer Engineering

Dr. Andres Kwasinski, Assistant Professor
Committee Member, Department of Computer Engineering

Dr. Dhireesha Kudithipudi, Assistant Professor
Committee Member, Department of Computer Engineering
Title:

CHAracterization of Relevant Attributes using Cyber
Trajectory Similarities

I, Jordan Bean, hereby grant permission to the Wallace Memorial Library to reproduce
my thesis in whole or part.

________________________________________
Jordan Bean

________________________________________
Date
This thesis is dedicated to my parents, family, friends and professors, who have always supported my work. To my parents Sam and Alison, my brother Alex, my sisters Mikayla and Carly, my thesis advisor Dr. Yang, friends Chris Murphy, Mike Snook, Josh Monsees, and all other family members, professors, friends and all-night lab companions who’ve helped me make it this far and whose list is too long to mention here.
Acknowledgments

The completion of this thesis was made possible by the never-ending encouragement of my parents Sam and Alison, the inspiration and dedication of Dr. Yang and the support of all my friends and family. Support and encouragement from my brother Alex, sisters Mikayla and Carly, friends Chris, Mike, Katie, and many others was also essential in the completion of my entire college career. A special thanks also goes to Rick Tolleson, and Charles Gruener, who have provided not only important information, materials, and knowledge but also friendship. Thank you to my many roommates and friends who both outside and inside the classroom made my college career a success.
Abstract

CHAracterization of Relevant Attributes using Cyber Trajectory Similarities

Jordan Bean

Supervising Professor: Dr. Shanchieh Jay Yang

On secure networks, even sophisticated cyber hackers must perform multiple steps to implement attacks on sensitive data and critical servers hidden behind layers of firewalls. Therefore, there is a need to study these attacks at a higher multi-stage level. Traditional taxonomy of cyber attacks focuses on analyzing the final stage and overall effects of an attack but, not the characteristics of an attack movement or ‘trajectory’ on a network. This work proposes to investigate trajectory similarities between multi-stage attacks, allowing for the characterization of both a hacker’s behavior and vulnerable attack paths within a network.

Currently, Intrusion Detection Systems (IDS) report alerts to a network analyst when a malicious activity is suspected to have occurred on a network. Previous work in this field has used IDS alerts as evidence of multi-stage attacks, and has thus been able to group correlated alerts into cyber attack tracks. The main contribution of this work is to use a revised Longest Common Subsequence (LCS) algorithm to analyze attack tracks as trajectories. This allows a systematic analysis to determine which alert attributes within a track are of great value to the characterization of multi-stage attacks. The basic LCS
algorithm, which looks for the longest common sequence in two strings of data, is extended to support the non-uniformity of alert data using a time windowing system. In addition, a normalization method will be applied to ensure that the attack track similarity measure is not adversely affected by differences in attack track length. By applying the revised LCS algorithm, attack trajectories defined in terms of various IDS alert attributes are analyzed. The results provide strong indicators of how multidimensional cyber attack trajectories can be used to differentiate attack tracks.
Contents

Dedication .................................................................................. iii

Acknowledgments ....................................................................... iv

Abstract ................................................................................... v

1 Introduction ............................................................................ 1
  1.1 Motivation ........................................................................ 1
  1.2 Multi-Stage Cyber Attack Tracks ...................................... 3
    1.2.1 Alert Correlation .................................................. 3
    1.2.2 Non-Uniform Alert Spacings ................................. 4
    1.2.3 Alert Attributes .................................................. 4
  1.3 Overview and Thesis Goals ............................................ 6
    1.3.1 Alert Attribute Analysis .................................... 7
    1.3.2 Matching of Time Stretched Attack Tracks ............ 8
    1.3.3 Increased Knowledge Basis for Prediction Using Multiple Alert Attributes .................................. 8

2 Related Work ......................................................................... 9
  2.1 Similarity Comparison with Subsequence Matching Algorithms .... 10
    2.1.1 Euclidian Distance .............................................. 10
    2.1.2 Principle Components Analysis Using Euclidian Distance ................................................. 10
    2.1.3 Hausdorff Distance ............................................ 11
    2.1.4 Hidden Markov Model-based Distance ................. 12
    2.1.5 Dynamic Time Warping ..................................... 13
    2.1.6 Longest Common Subsequence .................... 14
  2.2 An Alternative to Using Similarity value for Trajectory Characterization 16
  2.3 Capabilities and Limitations of Subsequence Matching and Shape Context Descriptors .................................................. 17
  2.4 Advantages of Longest Common Subsequence .................. 18
    2.4.1 Noise Reduction ................................................ 18
2.4.2 Handling of Time Stretched Attack Tracks .......................... 19

3 Methodology .................................................................................. 21
  3.1 Mapping of Alert Attributes .......................................................... 21
    3.1.1 Alert Severity ........................................................................ 21
    3.1.2 Destination IP Address ............................................................ 22
    3.1.3 Services Attacked ................................................................. 23
    3.1.4 Protocol ................................................................................. 26
  3.2 Adaptation and Enhancement of the Longest Common Subsequence Method 26
    3.2.1 Alert Non-Uniformity and Time Windowing ............................. 26
    3.2.2 Comparison Granularity ......................................................... 28
  3.3 Determination of Important Alert Attributes and Entropy Calculation .... 29

4 Results and Analysis ........................................................................ 36
  4.1 Network Simulator ....................................................................... 36
  4.2 Similarity Results ......................................................................... 37
  4.3 The Relationship of Entropy Based on Trajectory Similarity .......... 40
  4.4 Granularity Analysis of Attributes ................................................. 44
    4.4.1 Alert Severity ........................................................................ 45
    4.4.2 Destination IP Address/Subnet ............................................... 46
    4.4.3 Services Attacked ................................................................. 47
    4.4.4 Protocol Used ........................................................................ 48
  4.5 Network Topology ........................................................................ 48
  4.6 Efficiency and Stealth Factor ......................................................... 50
  4.7 Results ......................................................................................... 53

5 Conclusions ..................................................................................... 56
  5.1 Future Work ................................................................................ 57

Bibliography ...................................................................................... 59
# List of Tables

1.1 Alert attribute overview .................................................. 5

3.1 Damage scores assigned to exposures ............................. 22
3.2 Damage scores assigned to expanded exposures ............... 23
3.3 Grouping of Services Histograms ................................. 34
3.4 Sample similarity matrix .............................................. 35

4.1 An example 7-step cyber attack. ................................. 37
4.2 A similarity matrix for 15 attack tracks examined on the severity attribute. 38
4.3 Results of changing the granularity over a random assortment of attacks. 46
4.4 Random attacks over the increased breadth network .......... 49
4.5 Random attacks over the increased depth network .......... 49
4.6 Random attacks on a large network with low efficiency (0.1) and a high stealth (0.9) .............................................. 52
4.7 Entropy results over multiple attack scenarios ................. 54
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>A visual representation of an attack track as a 3-dimensional trajectory</td>
<td>6</td>
</tr>
<tr>
<td>2.1</td>
<td>From left to right; shell binning with 4 shells, sector binning with 8 sectors, and combined binning with 32 bins</td>
<td>16</td>
</tr>
<tr>
<td>2.2</td>
<td>Attack trajectories containing the same severity progression over a different time scale</td>
<td>20</td>
</tr>
<tr>
<td>3.1</td>
<td>Network Zones</td>
<td>24</td>
</tr>
<tr>
<td>3.2</td>
<td>A visual representation of an attack track as a 3-dimensional trajectory.</td>
<td>28</td>
</tr>
<tr>
<td>3.3</td>
<td>A graphical representation of the output entropy based on a single input similarity value.</td>
<td>32</td>
</tr>
<tr>
<td>4.1</td>
<td>Two trajectories that illustrate a basic similarity.</td>
<td>39</td>
</tr>
<tr>
<td>4.2</td>
<td>Two trajectories that illustrate a more complex similarity.</td>
<td>40</td>
</tr>
<tr>
<td>4.3</td>
<td>Two trajectories, one of which has been realigned to more clearly show similarity.</td>
<td>41</td>
</tr>
<tr>
<td>4.4</td>
<td>Two trajectories that have a low similarity with each other.</td>
<td>42</td>
</tr>
<tr>
<td>4.5</td>
<td>A histogram showing the varying values of similarity in Table 4.2.</td>
<td>43</td>
</tr>
<tr>
<td>4.6</td>
<td>Network with an increased breadth.</td>
<td>50</td>
</tr>
<tr>
<td>4.7</td>
<td>Network with an increased depth.</td>
<td>51</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

This work seeks to develop a methodology that extracts similarities between cyber attack movements within a network. These movements, also referred to as ‘trajectories’ or ‘tracks’ in the literature and this thesis, may be defined in terms of various attributes reported in intrusion alerts. Our study suggests the use of Longest Common Subsequence (LCS) to determine similarities between attack trajectories and adapts an entropy metric to evaluate the importance of individual alert attributes. The results of this thesis should provide a better knowledge base for a robust cyber attack track classification system as well as for a cyber alert prediction systems.

1.1 Motivation

Modern networks that contain large amounts of important and often confidential or sensitive information are generally kept on secure networks that use multiple layers of protection against cyber hackers. As a result, hackers must use sophisticated attacks that require many steps in order to infiltrate these multi-layer networks. The use of a multi-layered network approach is common because it allows for restrictions to be placed on traffic that flows into different portions of the network, especially those which are further away from direct Internet access [2]. This configuration allows for a much more secure network where more
sensitive data can be hidden deeper in a more protected location of the network.

As the Internet and cyber hacking become increasingly prevalent, the methods with which intruders are able to execute cyber attacks also become increasingly sophisticated. Moreover, many methods used by knowledgeable hackers are now being published as scripts that can be run by less experienced hackers [12]. This contributes directly to an increase in the number of sophisticated attacks that regularly occur on complex networks. Consequently, a mechanism which can recognize similarities among attacks that contain multiple steps would be directly suited towards this portion of the cyber security problem.

Hansman and Hunt [12] identify multiple categories of attacks that can occur on a network including: viruses, worms, buffer overflows, denial of service attacks, network attacks, physical attacks, password attacks, and information gathering attacks. This is one of the first works which classifies attacks based on the entire multiple step process that an attack takes over a period of time [12]. Furthermore, Hansman and Hunt stated that their solution aimed to create a classification which is not too general to be useful [12]. This is a common problem that many classification systems which study attacks at the multistep level suffer from. In the conclusion of their work, Hansman and Hunt determined that a method able to correlate attacks that contain multiple dimensions of data would be of great use [12]. Additionally, they concluded that an increased knowledge base in terms of multi-stage attacks would provide essential knowledge for the proper and complete satisfaction of all classification requirements [12]. The work done in this thesis will examine these challenges.
1.2 Multi-Stage Cyber Attack Tracks

The production of alerts by Intrusion Detection Systems (IDS) generally results in large data sets which quickly become unmanageable by a network analyst. For this reason, much work has been done in the way of alert correlation techniques [5]. As a result of the work done in [5] the notion of a multi-stage cyber attack track was created. In an attempt to reduce the complexity with which alert data is presented, alert correlation arranges IDS alerts into time ordered sequences. Each sequence represents a group of alerts that is suspected to have been generated by the actions of one or multiple hackers working together to reach a single goal. As an inherent result of the manner in which attacks are both performed by hackers and reported by IDS, the resulting attack tracks contain data that is non-uniformly spaced in time. In the sections that follow the ideas of alert correlation and non-uniform alert spacing will be discussed more thoroughly.

1.2.1 Alert Correlation

Fava et al. indicate that many different methods are currently used to perform alert correlation [5]. These methods include; probabilistic correlation [30], behavior based correlation [5], triggered event correlation [31], combinations of these events as well as others [5].

A completely probabilistic approach to alert correlation is presented by Valdes and Skinner in [30]. Valdes and Skinner present a two sensor system in which they are able to monitor misuse of TCP and then perform correlation based on Bayesian Statistics [30].

A second method proposed by Xu and Ning focuses on using what are called triggering events to perform correlation of IDS alerts [31]. Xu and Ning first perform a grouping
based on similar low level events that trigger the production of IDS alerts. Then using a clustering algorithm, the alerts are separated into clusters which are expected to represent alerts that are generated from the same attack [31].

1.2.2 Non-Uniform Alert Spacings

Throughout the process of alert generation, reporting and correlation, various factors lead to a spacing of alert data that is non-uniform over time. Non-uniformity over time means that intrusion alerts will occur at random points throughout the time scale instead of at predetermined intervals. As will become clear later on, this is a very important concept with respect to the comparison of two trajectories, especially in terms of lining trajectories up on the same time scale. Additionally, the use of different tactics by cyber hackers over time will generally result in cyber alerts that are spaced in a non-uniform manner. In work done by Dain and Cunningham in [8] they present alert data gathered from a cyber alert competition at DEF CON. The alert data gathered over multiple correlated attacks shows a clearly non-uniform pattern.

1.2.3 Alert Attributes

At each alert in an attack track, the following alert attributes exist; date and time of report, exploited service, protocol used, source IP address, destination IP address, source port, destination port, and alert severity. As a result of the characteristics and common uses of each of these attributes, only a certain set was focused on for this thesis. The source IP address was ignored in this case because techniques like IP spoofing, where a hacker’s source IP address is faked, are fairly common [14]. Because this is commonly used among cyber
hackers to conceal their identities, it proves to be a fairly unreliable measure of a hacker’s action and could lead to misleading results and conclusions. Additionally, the source and destination ports are also ignored for this thesis. Because the service alert attribute provides more information than the destination port, it is not worthwhile to consider this attribute [14]. For reasons similar to that of the source IP address, the source port can be an unreliable measure and is consequently ignored. An overview of alert attributes and their use is given in Table 1.1.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Used</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severity</td>
<td>Yes</td>
<td>Provides a measure of an alerts severity or impact</td>
</tr>
<tr>
<td>DateTime</td>
<td>Yes</td>
<td>Provides time ordered reference</td>
</tr>
<tr>
<td>Service Exploited</td>
<td>Yes</td>
<td>Provides information regarding hacker’s interest</td>
</tr>
<tr>
<td>Destination IP Address</td>
<td>Yes</td>
<td>Provides locality characteristic</td>
</tr>
<tr>
<td>Source IP Address</td>
<td>No</td>
<td>Unreliable and/or misleading</td>
</tr>
<tr>
<td>Source Port</td>
<td>No</td>
<td>Unreliable</td>
</tr>
<tr>
<td>Destination Port</td>
<td>No</td>
<td>Covered in more detail by service attribute</td>
</tr>
</tbody>
</table>

Table 1.1: Alert attribute overview

As a result of the categorical nature of alert attributes, there is a direct need for a way to map attributes to a ranked value for comparison. The method used to map categorical alert attributes to ranked numerical values was based on the characteristics of each alert attribute. Furthermore, this ranking is highly dependent on the possible categorical values that each attribute can take on. Many of the alert attributes that are examined are mapped to numerical values based on the criticality with which their categories can be associated. This notion of mapping attribute categories based on criticality or threat level is an idea that was proposed and used in Argauer’s thesis [2]. Additionally, it should be noted that by no means do these mappings represent the only possible method of assigning numerical values to the categorical alert attributes found in cyber data. Moreover, many different, yet logical
and acceptable alert attribute mappings could be used to assign numerical values to the various categories presented. The mappings used in this thesis represent the best possible assignments that could be made based on the author’s current knowledge and available resources. A complete and more detailed description of the mapping of each of the alert attributes is given in the methodology section that follows.

1.3 Overview and Thesis Goals

This thesis plans to treat attack tracks as trajectories that occur in a multi-dimensional Euclidian space. In this notion of attack trajectories, alert attributes will be mapped to ranked values which can be used to plot an attack track in a Euclidian space. A visual representation of an attack track as a trajectory, which uses three alert attributes to define Euclidian dimensions is depicted below in Figure 1.1.

![Figure 1.1: A visual representation of an attack track as a 3-dimensional trajectory](image)

In terms of this representation, the Longest Common Subsequence (LCS) method was found to provide the best overall ability to match similar trajectories. Additionally, the
LCS method was adapted to handle data sequences that are not uniformly spaced in time, data sequences that are stretched out in time, and to provide some level of noise reduction. Moreover, the abilities of the LCS algorithm will be used to find all possible cases of similarity that may exist between two trajectories. The LCS method will be used to analyze multiple cyber attack scenarios to generate a similarity matrix. The entropy of each alert attribute will then be calculated using the similarity matrix values. This measure of entropy will be used to determine which alert attributes contain the most interesting data and which contain the least interesting data. A major advantage of this method of discovery is that no preconceived notion of which attack tracks are similar has to be developed. The results should indicate which alert attributes are the most important and give the most reliable information when considering analysis of attack tracks. Overall, this will improve knowledge of attack track characteristics and could provide a more concrete basis for further alert prediction methods.

1.3.1 Alert Attribute Analysis

Because attack tracks are a relatively new concept, there is a need to determine their most important properties in terms of cyber hacker analysis. The overall goal of this work is to determine the most important and meaningful alert attributes that should be taken into account when performing attack track analysis. This will be accomplished using an algorithm which examines two attack tracks and determines how similar they are with respect to each other. The similarity of two attack tracks will be computed by comparing the values of one alert attribute throughout the two tracks. This similarity value will then be used in the calculation of the entropy value which will aid in the determination of the most important
alert attributes.

1.3.2 Matching of Time Stretched Attack Tracks

One of the main contributing factors to the spacing of IDS alert data in a non-uniform fashion over time is the fact that many hackers and many attacks are performed over varying periods of time. Attacks that occur over a period of time that is stretched out in comparison to other attacks fall into a special subset of non-uniform data sequences that can be classified as time stretched attacks. A study performed by Moitra and Konda in [24] examines data from CERT at the Carnegie Mellon Software Engineering Institute of attacks on various websites. One attribute that the study examines is the time between web attacks or incidents. Through this study Moitra and Konda found that attacks occur over periods of time as long as many days or as short as within a single day [24]. This thesis will address the problem of detecting similarities among stretched out time sequences by examining different data sequence comparison methods to determine which can best tackle this issue.

1.3.3 Increased Knowledge Basis for Prediction Using Multiple Alert Attributes

The overall importance of this work is laying a foundation for improvement in future works. In particular, one area that should benefit greatly from an increased knowledge of cyber attack track characteristics is the field of alert prediction. By determining which attributes best define the characteristics of an attack track, prediction methods should have a more complete basis for their work and can hopefully increase prediction accuracy. Many methods, including those presented in [5], [15], and [6] as well as others perform cyber attack prediction.
Chapter 2

Related Work

For this thesis the main analysis method chosen was that of subsequence matching. Since attack tracks are essentially time ordered sequences of data, subsequence matching methods are a very attractive analysis option. Wu et al. conclude that the use of subsequence matching techniques on data that occur in the form of time series can lead to important insight in regards to the overall cyber attack problem domain [11]. Furthermore, Wu et al. state that subsequence similarity matching can be used to improve the areas of prediction and rule discovery by providing insight into the structure of time series data [11]. Many areas outside of the cyber domain have used subsequence matching for the purpose of performing analysis on time series data including; analysis of the stock market, weather forecasts and predictions, process workloads, budget predictions, as well as other applications [11]. The application of a subsequence matching method to the cyber domain, particularly the field of attack track analysis, should provide much greater insight into the overall structure and characteristics of cyber attacks.
2.1 Similarity Comparison with Subsequence Matching Algorithms

2.1.1 Euclidean Distance

The standard Euclidean Distance algorithm is used to find the straight line distance between two points in a Euclidean space and is defined in the equation given below [3].

\[ D(A, B) = \sqrt{(a_x^n - b_x^n)^2 + (a_y^n - b_y^n)^2} \]  

By simply taking the average of the Euclidean distance between all points in two trajectories or sequences of data, a measure of their similarity can be obtained. This is represented formally in the equation given below [32]

\[ D(A, B) = \frac{1}{N} \sum_{n=1}^{N} [(a_x^n - b_x^n)^2 + (a_y^n - b_y^n)^2]^{\frac{1}{2}} \]

This idea of performing a point wise comparison is a central concept to many similarity comparison methods. Because of its relative simplicity, this measurement of similarity by itself does not represent the best cyber attack trajectory comparison method.

2.1.2 Principle Components Analysis Using Euclidean Distance

A second measure of similarity studied by Zhange et al. in [32] was a combination of Principle Components Analysis (PCA) and Euclidean Distance measurement that was developed by Bashir et al. in [9]. The work done by Bashir et al. was designed for application in trajectories found in video data. Preprocessing of video data was done using a Karhunen-Loeve Transform (KLT) for the compression of a 1-Dimensional signal representation of video data trajectories [9]. Noise reduction was then performed on the resulting signal
using a wavelet transform called DB4 [9]. Using coefficients derived from the wavelet transform, a matrix is formed that represents the trajectory data. Single Value Decomposition (SVD) is then applied to obtain the principle trajectory components [9]. By using the Euclidian Distance, a measure of distance between the PCA components of the two trajectories was found. Let the values $E_{Am}$ and $E_{Bm}$ represent the distance measure of the $m^{th}$ component of trajectories $A$ and $B$ respectively and the similarity value was calculated using Equation (2.3) [9].

$$D(A, B) = \sum_{m=1}^{M} [(E_{Am} + E_{Bm})^2]^{\frac{1}{2}}$$  \hspace{1cm} (2.3)

### 2.1.3 Hausdorff Distance

The use of the Hausdorff distance as a measure of similarity was also considered in [32]. This distance measure is calculated using the following set of equations [17]

$$D_c = \min_{A,B}\{D_{A,B}, D_{B,A}\}$$  \hspace{1cm} (2.4)

$$D_{A,B} = \max_{i=0\ldots T}\{\min_{j=0\ldots T}(d_{i,j})\}$$

$$D_{B,A} = \max_{i=0\ldots T}\{\min_{j=0\ldots T}(d_{i,j})\}$$

where $A$ and $B$ are two trajectories being compared for similarity and $d_{i,j}$ is the Euclidian distance measure between points $i$ and $j$ on two different trajectories. A similarity metric was then computed by calculating the difference in the speed of the points on one trajectory with the speed of the points on a second trajectory [17].
2.1.4 Hidden Markov Model-based Distance

In work done by Porikli et al., a Hidden Markov Model (HMM) based approach is used to examine the various features used in the processing of video data [26]. Porikli et al. used the notion of trajectories in video data to represent the various movements of objects throughout the course of a video [26]. Similar to the manner in which the alert attributes are used to make up attack track trajectories, Porikli used the properties of movement during the course of a video to define the various features that make up video data trajectories [26].

The features used to define video trajectories in [26] were: speed, orientation, location, size, and aspect ratio. These features play an important role in the detection of similarities among video data trajectories. Porikli et al. first define a similarity measurement between two objects \( j \) and \( k \) using Equation (2.5) given below.

\[
m_{jk} = e^{-d(j,k)/2\sigma^2}
\]

(2.5)

where \( d(j, k) \) is a measure of distance and \( \sigma^2 \) is used as a constant scaling value [26]. The distance measure is calculated using a likelihood function and the HMM representation of the trajectories being compared. The resulting distance measure for two trajectories \( T^a \) and \( T^b \) is formally presented below in Equation (2.6) [26].

\[
d(T^a, T^b) = | L(T^a; \lambda^a) + L(T^b; \lambda^b) - L(T^a; \lambda^b) - L(T^b; \lambda^a) |
\]

(2.6)

where \( \lambda^a \) and \( \lambda^b \) are the HMMs of trajectories \( a \) and \( b \) respectively and \( L() \) represents a likelihood function [26]. Using this similarity measure, Porikli et al. were able to create an
affinity matrix $M_{jk}$ which represents the similarity of each of the objects $j$ and $k$ [26].

### 2.1.5 Dynamic Time Warping

The Dynamic Time Warping (DTW) algorithm is based on the idea that when comparing two sequences of time series data that they must be aligned appropriately first [19]. Moreover, this algorithm performs a warping, a stretched mapping, from data points on one sequence to data points on a second sequence. However, the use of DTW can result in many points in one sequence being mapped to a single point in another sequence. This singularity problem develops an unexpected and undesired result in the alignment of two data sequences [19].

Consider the following data sequences

$$A = a_1, a_2, \ldots, a_n \quad (2.7)$$

$$B = b_1, b_2, \ldots, b_m \quad (2.8)$$

where data sequence $A$ is of length $n$ and data sequence $B$ is of length $m$. A matrix of size $n$-by-$m$ that holds the distance between every possible set of points in the two sequences, $A$ and $B$, is then constructed. In General, the Euclidian distance is used for this measurement. The similarity of two data sequences is then calculated by finding the continuous path from one corner of the matrix to the opposite corner of the matrix that minimizes the total cost [19]. This is referred to as finding the warping path that minimizes the overall warping cost [19]. The minimal warping cost is calculated as
\[ DTW(A, B) = \min \left\{ \frac{\sum_{k=1}^{K} w_k}{K} \right\} \]  

(2.9)

where \( w_k \) is the \( k^{th} \) element in the warping path, and \( K \) is a compensation factor used to account for sequences of different lengths [19] [32].

Keogh and Pazzani in [19], give the equation for finding the warping path as

\[ \gamma(i, j) = d(a_i, b_j) + \min\{\gamma(i - 1, j - 1), \gamma(i - 1, j), \gamma(i, j - 1)\} \]  

(2.10)

where \( \gamma(i, j) \) is the cumulative distance. This cumulative distance is defined as the distance of the current cell plus the minimum cumulative distance of the adjacent cells [19].

### 2.1.6 Longest Common Subsequence

The standard LCS algorithm looks for consecutive similarities in two sequences of data starting at one end and moving to the other. Let T1 and T2 be attack tracks as defined by the simulation data where \( i \) and \( j \) are the size of attack tracks T1 and T2 respectively. The alerts in each track, T1 and T2, can be defined as \( a_1, a_2, ... a_i \) and \( b_1, b_2, ... b_j \). The LCS algorithm can thus be defined in a recursive fashion [23] as
$$LCS(T1[i], T2[j]) =$$

$$\begin{cases} 
0 & \text{if } i = 0 \text{ or } j = 0 \\
1 + LCS(T1[i - 1], T2[j - 1]) & \text{if } i, j > 0 \text{ and } a_i = a_j \\
\max(LCS(T1[i - 1], T2[j]), LCS(T1[i], T2[j - 1])) & \text{otherwise}
\end{cases}$$

(2.11)

The result of a comparison between two attack tracks, $T1$ and $T2$, is the largest number of consecutive points with the same value. For example, let an attack track $T1$ be of length $m$, then the result of $LCS(T1, T1)$ will be the value $m$. Similarly, let an attack track $T2$ be of length $n$, where $n < m$, then the result of $LCS(T2, T2)$ will be the value $n$. This creates an obvious problem in that the similarity value resulting from the two LCS comparisons shows that $T1$ has a higher similarity value with itself than $T2$ has with itself. To provide a better metric for comparison a normalization method was introduced. The calculation of the normalized similarity measure is shown formally in the equation below [23].

$$S(T1, T2) = \frac{LCS(T1, T2)}{\min(|T1|, |T2|)}$$

(2.12)

In the normalization equation, $|T|$ represents the number of alerts in an attack track. All of the resulting similarity values will fall within the range of [0.0, 1.0]. Using this notion of similarity, the results of the two comparison examples, $LCS(T1, T1)$ and $LCS(T2, T2)$, will be of equal value.
2.2 An Alternative to Using Similarity value for Trajectory Characterization

In the work presented by Kholgade [20] a two-dimensional shape context algorithm is extended to incorporate characterization of an object’s speed and direction of motion. The motion classification presented by Kholgade relies on an initial shape context analysis. This method extends the idea of a two-dimensional shape context to a third and fourth dimension defined by an object’s speed and direction of motion [20]. Kortgen et al. in [22] define a method that uses binned histograms to create a two-dimensional shape context. The two-dimensional model used to explain this shape context notion consists of three parts (Figure 2.1); a shell binning model, a sector binning model and a combined shell and sector binning model [20, 22].

![Figure 2.1: From left to right; shell binning with 4 shells, sector binning with 8 sectors, and combined binning with 32 bins](image)

In this notion, the shell binning model is used to define a single dimension (e.g. the x-dimension), the sector binning model is used to define a second dimension (e.g. the y-dimension), and the combined binning model is used to provide an overall shape context. In [20], Kholgade introduces the idea of extending this shape context model to take an object’s motion attributes into account. Kholgade’s model includes an object’s speed as a
third dimension in the shape context model and an object’s direction of motion as a fourth
dimension. Using this enhanced model Kholgade is able to provide not only shape context
but, an object’s characterization through motion context as well.

2.3 Capabilities and Limitations of Subsequence Matching and Shape Context Descriptors

Much of the work studied by Zhang et al. is aimed specifically at the application of process-
ing video surveillance data [32]. The major overall conclusion that Zhang et al. realized
was that all of the methods other than DTW and LCS are generally fitted to working on
problems that examine differences in position similarity. Zhang et al. discovered that al-
though the Euclidian Distance measure and the PCA combined with Euclidian Distance
measure were significantly faster at processing data than the LCS and DTW methods, they
were less accurate overall [32]. Moreover, it was discovered that the Euclidian Distance
and PCA combined with Euclidian Distance measure suffered more easily from noise that
occurred in the time dimension [32]. However, overall Zhang et al. determined that the
DTW and LCS methods were better suited towards performing matching that focuses on
shape similarity and not the position based similarities that they were looking for in video
surveillance data [32]. Furthermore, they found the HMM and Hasudorf distance measures
to be poor in many aspects examined and thus not very useful for similarity detection in
trajectories. The HMM model was found to have a problem with over fitting of models as
a result of the small training set that is available [32]. The Hasudorf distance was found to
have trouble recognizing the difference between two trajectories as a result of the pair wise
point comparison method that it uses [32].
2.4 Advantages of Longest Common Subsequence

Huang et al. [32] analyzed multiple sequence comparison methods for finding similarities in data sequences. Based on their results, they concluded that LCS is best suited for finding any possibly shape similarity that could occur in sequences of data. A major reason for this is the ability of LCS to line trajectories up appropriately and to be able to handle noise that occurs in the time dimension [32]. This second ability is extremely important considering the non-uniform nature of the cyber alert trajectory data. In terms of the cyber attack track application domain, this is an advantage. This is a direct result of being able to visualize cyber attack tracks as trajectories in free space, where the shape of each trajectory is defined by its alert attributes. Considering this characterization the search for shape similarity in trajectories is directly applicable to the problem of finding similar cyber attack tracks.

Additionally, the new trend in which novice hackers use published scripts to employ sophisticated attacks has become increasingly more common. However, these types of attacks may be recognized quite easily by the LCS method due to the fact that use of the same script by a hacker should generate similar resulting IDS alert sequences. This should prove to be a great advantage of the LCS method over many of the other currently existing methods.

2.4.1 Noise Reduction

Although Huang et al. [32] claim that both DTW and LCS work best on problems that look for similarity of trajectory position or shape, Gunopulos et al. [23] claim that LCS has an advantage over the DTW method in terms of noise rejection. This is due in large
part to the fact that DTW requires all the elements in a sequence including the outliers to be matched, while the LCS method allows for some mismatch of data points [23]. This is an important issue because cyber attack tracks may have inherent missing or noisy data due to correlation error or sophisticated and stealthy attacks. An alert provides a snapshot or a single piece of evidence within an entire attack track. IDS are not guaranteed to capture all of the possible alerts that occur during a multi-stage attack and therefore, the possibility of missing alerts must be taken into account. Additionally, normal network communication and advanced cyber hacking methods can create alerts that are unrelated to any end goal which, can in turn create noise in an attack track.

2.4.2 Handling of Time Stretched Attack Tracks

Due to the details of its implementation, the LCS method is able to detect similarities among attack tracks that are stretched out over different periods of time. An example of a comparison involving a time stretched attack is shown in Figure 2.2 where only two dimensions, time and attack severity are being considered.

As can be seen in Figure 2.2 attack Trajectory 1 and attack Trajectory 2 contain the exact same progression of attack severity only over a very different time scale. Due to the nature of even the basic LCS algorithm, these trajectories will be detected as exactly the same.
Figure 2.2: Attack trajectories containing the same severity progression over a different time scale
Chapter 3

Methodology

As a result of its many advantages in terms of the cyber attack application domain, the Longest Common Subsequence (LCS) method was used for similarity detection in this thesis [23]. Furthermore, the similarity measure provided by the LCS method was used in conjunction with a measure of entropy to accomplish the reduction of cyber attack track data dimensionality [21]. The combination of these two methods ensures that attack track progression as a trajectory is taken into account during the discovery of the important of alert attributes.

3.1 Mapping of Alert Attributes

3.1.1 Alert Severity

The notion of a severity attribute in a cyber alert was developed by Argauer in [2] for the purpose of associating a specific exposure level with alerts. This exposure assignment is performed using the Virtual Terrain Assisted Impact Assessment for Cyber Attack Tracks (VTAC) processing that is completed during the correlation of alerts into attack tracks. In Argauer’s thesis, a damage score is assigned to each exposure as a measure of the severity that may result [2]. Additionally, gaps were left between the different mapped values to account for differences in privilege levels [2]. The damage scores that Argauer developed
are given in table 3.1 below.

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>Damage Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reconnaissance</td>
<td>1</td>
</tr>
<tr>
<td>Intrusion (User)</td>
<td>4</td>
</tr>
<tr>
<td>Intrusion (System)</td>
<td>7</td>
</tr>
<tr>
<td>Escalation</td>
<td>7</td>
</tr>
<tr>
<td>Goal</td>
<td>10</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3.1: Damage scores assigned to exposures

In order to accommodate an increasingly detailed list of exposures, the severity attributes and damage scores associated with them was expanded. The expanded categories and their associated damage scores are given in Table 3.2.

In this instance the basic rule of thumb was followed in which categories with a higher threat level or criticality were mapped to higher numerical values. The mappings performed for the expanded list of severity categories was originally developed by Haitao Du \(^1\) and then later verified by Jared Holsopple a known expert on the subject.

### 3.1.2 Destination IP Address

For this thesis the destination IP address was used as a characterization of a hacker’s movement throughout a network. Accordingly, a simulation network was divided up into its individual subnets for similarity comparison. This is a direct application of the work done in [2] where a layered network approach was used to differentiate between parts of a network that were configured in different manners. Moreover, this division of a simulation network into zones allows a hacker’s movement throughout a network to be used for the purpose of characterization based on the different network layers that have been accessed.

\(^1\)Conversation with Haitao Du, Network and Information Processing lab.
<table>
<thead>
<tr>
<th>Attack Type</th>
<th>Damage Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reconnaissance Scanning</td>
<td>1</td>
</tr>
<tr>
<td>Reconnaissance Enumeration</td>
<td>5</td>
</tr>
<tr>
<td>Reconnaissance Footprinting</td>
<td>5</td>
</tr>
<tr>
<td>Reconnaissance Sniffing</td>
<td>5</td>
</tr>
<tr>
<td>Reconnaissance Other</td>
<td>5</td>
</tr>
<tr>
<td>Reconnaissance Undetected</td>
<td>8</td>
</tr>
<tr>
<td>Miscellaneous Other</td>
<td>11</td>
</tr>
<tr>
<td>Miscellaneous Undetected</td>
<td>12</td>
</tr>
<tr>
<td>Miscellaneous VirusTrojan</td>
<td>15</td>
</tr>
<tr>
<td>Escalation Service</td>
<td>22</td>
</tr>
<tr>
<td>Escalation Undetected</td>
<td>27</td>
</tr>
<tr>
<td>Escalation OS</td>
<td>28</td>
</tr>
<tr>
<td>Intrusion Other</td>
<td>30</td>
</tr>
<tr>
<td>Intrusion User</td>
<td>32</td>
</tr>
<tr>
<td>Intrusion Undetected</td>
<td>37</td>
</tr>
<tr>
<td>Intrusion Root</td>
<td>38</td>
</tr>
<tr>
<td>Goal Dos</td>
<td>40</td>
</tr>
<tr>
<td>Goal Other</td>
<td>41</td>
</tr>
<tr>
<td>Goal Backdoor</td>
<td>42</td>
</tr>
<tr>
<td>Goal Espionage</td>
<td>44</td>
</tr>
<tr>
<td>Goal Pilfering</td>
<td>44</td>
</tr>
<tr>
<td>Goal Corruption</td>
<td>44</td>
</tr>
<tr>
<td>Goal Undetected</td>
<td>44</td>
</tr>
</tbody>
</table>

Table 3.2: Damage scores assigned to expanded exposures

The simulation network created and the corresponding network zones are presented in Figure 3.1.

3.1.3 Services Attacked

The network services attribute represents the services that are exploited throughout an attack by a cyber hacker. The exploitation of network services throughout the course of an attack track will be monitored using a modified histogram to keep track of which services are attacked most often. The modified histogram consists of five groups or five separate
The groupings presented in Table 3.3 were developed using the same basic rule which
uses a measure of criticality as a mapping guide. In this instance, the services were entered into groups based on criticality. Using background information from [1], [18], [7], [28], [10], [13], [16], [27], [29], [33] the overall determination for each of the original groupings was made. Additionally, data from work done by Argauer was used to support these resultant groupings [2]. For example, Argauer noted that both the services FTP and DNS had relatively high impacts associated with their being compromised and so they were placed in group 2 to reflect their relatively similar criticalities. Furthermore, research on these two services and prior knowledge of the importance of these services led to the verification that group 2 was an acceptable place for these particular services to reside. It should be noted again that the groupings formed here are somewhat representative of opinion and are most certainly subject to interpretation. Multiple groupings that are each justifiable in their own right are certainly more than possible and in some cases may be the result of a network’s setup, especially in real world applications.

The actual mapped values were determined by keeping a histogram of which services were attacked most often. The actual number of times that a service was attacked was then used to compute the corresponding percentage of attacks that occur on that service in its particular group. For example, if in group 3 ICMP, IIS, and IMAP were all attacked 4 times and FPSE and LotusServer were attack 6 times then the resulting percentages for each of ICMP, IIS and IMAP would be around 16.7% and for FPSE and LotusServer the percentages would be 25%. This percentage was then multiplied by 10 and then the lowest possible base value in the corresponding group was added to the result to complete each service mapping. In the previously given example, the ICMP, IIS, and IMAP services would
all result in a value of approximately 21.7 calculated using the expression \((10 \times 16.7) + 20 = 21.7\). In this manner, each service was mapped to a numerical value within its own predefined group.

3.1.4 Protocol

The protocol attribute of a cyber alert was compared in a categorical fashion. Moreover, the protocol categories were only noted as being similar if the two categorical values were exactly the same. Because there is only a small number of protocols that are used for transmission over the internet, and because there is very little evidence to suggest major vulnerability or exposure differences between the protocols they will most likely be kept as categorical values for this thesis.

3.2 Adaptation and Enhancement of the Longest Common Subsequence Method

3.2.1 Alert Non-Uniformity and Time Windowing

Due to the non-uniform nature of the attack track alert data, some enhancements were required in order to make the LCS algorithm work as desired. A major overall goal of these enhancements was to ensure that any possible similarities that exist between trajectories is found. Moreover, these adjustments were made to the basic LCS algorithm to provide a solution that was better suited to the detection of attack trajectory similarity. To increase the effectiveness of the LCS comparison method, the alert data was first normalized in time. Normalization was accomplished by setting the first alert in an attack track to time zero and then using the time stamp attribute to appropriately shift the following alerts to maintain
the correct time distance between alerts. Furthermore, a matching threshold was added to the basic LCS algorithm in place of the requirement for an exact match when performing comparison of alert attribute values. In essence, the basic LCS algorithm was changed to the form presented in Equation (3.1) where $\alpha$ represents the matching threshold [23].

$$LCS_{\alpha}(T_1[i], T_2[j]) =$$

\[
\begin{cases}
0 & \text{if } i = 0 \text{ or } j = 0 \\
1 + LCS_{\alpha}(T_1[i - 1], T_2[j - 1]) & \text{if } i,j > 0 \text{ and } \text{abs}(a_i - a_j) > \alpha \\
\max(LCS_{\alpha}(T_1[i - 1], T_2[j]), LCS_{\alpha}(T_1[i], T_2[j - 1])) & \text{Otherwise}
\end{cases}
\]

(3.1)

The inherently non-uniform alert data was handled by implementing a time windowing system. When a comparison resulting in a difference value outside of the matching threshold is encountered, the time windowing system will be invoked. The time windowing system allows the point that is latest in time to remain as a fixed comparison point while the second comparison point is updated to a later point in time. This is illustrated by examining the two attack tracks presented in figure 3.2.

![Comparison #1](image1.png)  ![Time windowing invoked](image2.png)

(a) First comparison  (b) Second comparison, time windowing invoked
Let the points in track 1 be defined by data points \( a_1, a_2, \ldots, a_i \) and the points in track 2 be defined by data points \( b_1, b_2, \ldots, b_j \). Let the result of the first comparison, between points \( a_1 \) and \( b_1 \) fall within the matching threshold. The second comparison, between data points \( a_2 \) and \( b_2 \) will result in a difference value that is not within the matching threshold. At this point the time windowing system will take into account that data point \( b_2 \) is later in time than data point \( a_2 \), and will perform a comparison of data points \( b_2 \) and \( a_3 \). Comparison of data points \( b_2 \) and \( a_3 \) will also result in a difference value outside of the matching threshold. Because data point \( b_2 \) is still later in time than data point \( a_3 \), the time windowing system will take effect and the data points \( b_2 \) and \( a_4 \) will be compared. The comparison of these two data points will result in a difference value that is within the matching threshold and the LCS algorithm will continue.

### 3.2.2 Comparison Granularity

The factor \( \alpha \) introduced in equation (3.1) is an adjustable parameter that can change the granularity of each similarity comparison made in the LCS algorithm. In their work Vlachos et al. refer to this adjustable parameter as a matching threshold [23]. Adjustments
were made to this parameter and the LCS algorithm was run over multiple data sets. The results were then analyzed to examine the effect of changes in the granularity.

3.3 Determination of Important Alert Attributes and Entropy Calculation

The resulting analysis of each of the alert attribute dimensions using the LCS algorithm provides measurements of similarity in the form of a similarity matrix. A sample similarity matrix created using five trajectories is given below in Table 3.4.

This matrix consists of values ranging from 0.0, indicating no similarity between two trajectories, to 1.0, indicating a completely matching set of trajectories. During the analysis of similarity matrix data, the diagonal was ignored because it represents comparisons of trajectories with themselves which, always match completely. Additionally, due to the implementation specifics of LCS, the similarity matrix will be symmetric with respect to the diagonal. Thus, only either the upper or lower triangular half of the matrix must be considered during analysis of a single alert attribute. The actual numerical values that occur within the similarity matrix for a single attribute indicate what percentage of two trajectories match. In the example given in Table 3.4 Trajectory 3 and Trajectory 2 are %60 similar to each other with respect to the shorter sequence. Later, it will be shown how the variation of similarity values present in the lower triangular half of the matrix will be used to determine the importance of the attribute represented. More specifically, it will be explained how the similarity percentages in the lower triangular half of the amtrix will be used to compute an entropy value which can then be used to discern attribute importance.
Determination of important alert attributes was accomplished using a measure of entropy introduced by Dash et al. in [21]. In an experimental study that they performed, Dash et al. used this measure of entropy as a replacement for Principle Components Analysis (PCA) for the purpose of discovering the most important dimensions in a data set [21]. Dash et al. found that use of the PCA algorithm required knowledge of original features which in some cases may not be possible. More specifically, during examination of certain data sets, Dash et al. found that many factors including large data sets and no prior knowledge of principle components contributes to a decrease in the effectiveness of PCA. As a result, they determined that entropy could be used to determine the importance that a dimension held based on the similarities that occurred between points within this dimension. This measure of entropy is formally defined in Equation (3.2) below.

\[
E = -\sum_{i=1}^{N} \sum_{j=1}^{N} (S_{ij} \times \log_2(S_{ij}) + (1 - S_{ij}) \times \log_2(1 - S_{ij})) \quad (3.2)
\]

where \(S_{ij}\) is Dash et al.’s measure of similarity between two trajectories [21]. This measure of similarity is defined where a value of 0.0 indicates pairs of trajectories that are very similar and a value of 1.0 indicates pairs of trajectories that have little to no similarity [21]. Because this represents an inverse relationship to the similarity value given by the LCS method, the values from this similarity matrix should be adjusted using the simple equation \(S_{ij} = 1 - Ls_{ij}\), where \(Ls_{ij}\) represents LCS similarities. However, closer examination of the entropy calculation shows that because both \(S_{ij}\) and \(1 - S_{ij}\) are used, the results of the inverse similarity values should match those of the LCS similarities taken directly. It also must be noted that since the entropy equation uses the base two logarithm, the similarity
values entered must be in between 1.0 and 0.0 not inclusive or (0.0, 1.0). Because the LCS method allows for similarity values that are between 1.0 and 0.0 inclusive or [0.0,1.0] simple heuristic measures had to be applied to values of 1.0 and 0.0. For LCS similarity values that fell on either the 1.0 or 0.0 bounds, the heuristics given in Equation (3.3) were used.

\[
S_{ij} = \begin{cases} 
\frac{1}{1+n^2} & \text{if } L_{s_{ij}} = 0.0 \\
\frac{n^2}{1+n^2} & \text{if } L_{s_{ij}} = 1.0 \\
1 - L_{s_{ij}} & \text{Otherwise}
\end{cases}
\]  

(3.3)

where \( n \) is the size of the shorter trajectory. The resulting entropy values were used to determine which attributes contain the most interesting and important data. This thesis defines the most important trajectories as those that are neither completely dissimilar or completely the same. Dash et al. define the entropy to indicate that trajectory pairs are either very close or very distant when the value of a single entropy calculation is close to 0.0 and that a pair of trajectories is close to the mean distance or mean similarity when the value for a single entropy calculation is close to 1.0 [21]. As a result, the most interesting data will be found in the attributes that have a higher overall entropy value over their entire similarity matrix.

To obtain a better understanding of what this entropy value indicates about a corresponding input data set, it was first graphed. The results of graphing the entropy equation from (3.2) are shown below in Figure 3.3.

Examination of this graph quickly shows that any resulting low entropy values will be
Figure 3.3: A graphical representation of the output entropy based on a single input similarity value.

As was previously discussed, the level of importance of trajectories can be determined by the amount of similarity between them. Many trajectories have great importance however, those trajectories which are complete matches or complete non-matches are of little interest. Revisiting Figure 3.3, it can be seen that only the trajectories which have complete matches or very high similarities and the trajectories that don’t match at all or have very low similarities will result in low entropy values. Since this is exactly what is desired based on the relationship between similarity and importance, this measure of entropy provides a
good overall characterization of the importance of alert attributes.
<table>
<thead>
<tr>
<th>Group</th>
<th>Services</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>ColdFusion, CVS, MS-SQL, MySQL, Oracle</td>
<td>40–50</td>
</tr>
<tr>
<td>Group 2</td>
<td>DNS, Finger, Frontpage, Frontpage 98, FTP</td>
<td>30–40</td>
</tr>
<tr>
<td>Group 3</td>
<td>ICMP, IIS, IMAP, FPSE, LotusServer</td>
<td>20–30</td>
</tr>
<tr>
<td>Group 4</td>
<td>NETBIOS, NFS, NNTP, PHP, phpBB</td>
<td>10–20</td>
</tr>
<tr>
<td>Group 5</td>
<td>POP3, Quicktime, RPC, SMTP, SNMP, SSH, Telnet, TFTP, UNIX, UPnP, Web CGI, Apache, Apache Tomcat, Oracle Web App Server, Microsoft Office, X Windows, XFS</td>
<td>0–10</td>
</tr>
</tbody>
</table>

Table 3.3: Grouping of Services Histograms
Table 3.4: Sample similarity matrix

<table>
<thead>
<tr>
<th></th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>1.0</td>
<td>0.25</td>
<td>0.3</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>T2</td>
<td>0.25</td>
<td>1.0</td>
<td>0.6</td>
<td>0.75</td>
<td>0.4</td>
</tr>
<tr>
<td>T3</td>
<td>0.3</td>
<td>0.6</td>
<td>1.0</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>T4</td>
<td>0.2</td>
<td>0.75</td>
<td>0.5</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>T5</td>
<td>0.8</td>
<td>0.4</td>
<td>0.5</td>
<td>0.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Chapter 4

Results and Analysis

4.1 Network Simulator

The data used throughout this paper was generated using a network simulation tool created by Costantini et al. [4]. This simulation tool was used to create a computer network based largely on the layered network approach introduced by Argauer in [2]. This approach is centered on the idea that data flowing into the network, in the direction away from the internet, should have limitations while the data that is heading out towards the internet can flow freely [2]. Additionally, the information flowing out of the given network should not be allowed to access the servers that are setup in what Argauer defines as a demilitarized zone that is directly connected to the internet [2]. The network developed for this work is shown above in Figure 3.1.

In this simulation network, various zones were established using the subnet addresses as a separation criterion. Zone 1 houses the servers that are directly connected to the internet and thus makeup the demilitarized section of the network, as was defined in Argauer’s layered network approach. Zone 2 houses internal servers that should be accessible to the users in zones 3 through 8 and can be thought of as a sort of intranet for these users [2]. The remaining zones makeup user occupied areas of the network that offer varying network
services.

Data from a single attack from a run of the simulation on a developed network is shown in Table 4.1.

<table>
<thead>
<tr>
<th>Step</th>
<th>Source IP</th>
<th>Target IP</th>
<th>(Port#) Service Exposure Description</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.5.231.72</td>
<td>192.168.1.5</td>
<td>(#80) WEB-MISC Webtrends HTTP probe</td>
<td>Discovered</td>
</tr>
<tr>
<td>2</td>
<td>252.10.175.172</td>
<td>192.168.1.5</td>
<td>(#80) WEB-IIS unicode directory traversal attempt</td>
<td>Compromised</td>
</tr>
<tr>
<td>3</td>
<td>192.168.1.5</td>
<td>192.168.2.3</td>
<td>(#80) WEB-IIS postinfo.asp access</td>
<td>Compromised</td>
</tr>
<tr>
<td>4</td>
<td>192.168.1.5</td>
<td>192.168.2.2</td>
<td>(#111) RPC mountd TCP exportall request</td>
<td>Compromised</td>
</tr>
<tr>
<td>5</td>
<td>192.168.1.5</td>
<td>192.168.2.3</td>
<td>(#80) WEB-IIS cmd32.exe access</td>
<td>Compromised</td>
</tr>
<tr>
<td>6</td>
<td>192.168.2.3</td>
<td>192.168.7.52</td>
<td>(#80) WEB-IIS cmd32.exe access</td>
<td>Compromised</td>
</tr>
<tr>
<td>7</td>
<td>192.168.7.52</td>
<td>192.168.8.14</td>
<td>(#20) FTP RNFR ./. attempt</td>
<td>Compromised</td>
</tr>
</tbody>
</table>

Table 4.1: An example 7-step cyber attack.

It should be noted that due to the form in which IDS sensors are arranged and report data, that repeat IDS alerts are fairly common in many systems. For the purposes of this work, the duplicate alerts have been removed to ensure a fair weighting and comparison scheme.

The LCS algorithm was verified using multiple test cases developed with Costantini’s simulator. This was accomplished with a manual attack generation capability that is available in the simulation tool [4]. Initially, trivial cases consisting of very similar or exactly the same attacks were tested. Then cases in which little or no matching should result were tested. After these trivial cases were verified as working correctly, more complex cases involving the use of time stretched attacks and attacks containing largely different alert spacings were tested. This effectively allowed for the verification of the time windowing capabilities.

### 4.2 Similarity Results

The actual result of an LCS comparison, in similarity matrix form, is shown in Table 4.2.
### Table 4.2: A similarity matrix for 15 attack tracks examined on the severity attribute.

<table>
<thead>
<tr>
<th></th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>T6</th>
<th>T7</th>
<th>T8</th>
<th>T9</th>
<th>T10</th>
<th>T11</th>
<th>T12</th>
<th>T13</th>
<th>T14</th>
<th>T15</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>1.0</td>
<td>0.6</td>
<td>0.4</td>
<td>1.0</td>
<td>0.6</td>
<td>0.4</td>
<td>1.0</td>
<td>0.2</td>
<td>0.6</td>
<td>1.0</td>
<td>0.2</td>
<td>0.6</td>
<td>1.0</td>
<td>0.6</td>
<td>0.4</td>
</tr>
<tr>
<td>T2</td>
<td>0.6</td>
<td>1.0</td>
<td>0.4</td>
<td>0.6</td>
<td>1.0</td>
<td>0.8</td>
<td>0.6</td>
<td>0.2</td>
<td>0.4</td>
<td>0.2</td>
<td>0.4</td>
<td>0.2</td>
<td>0.4</td>
<td>0.2</td>
<td>1.0</td>
</tr>
<tr>
<td>T3</td>
<td>0.4</td>
<td>0.4</td>
<td>1.0</td>
<td>0.4</td>
<td>0.4</td>
<td>1.0</td>
<td>0.4</td>
<td>0.2</td>
<td>0.71</td>
<td>0.4</td>
<td>0.2</td>
<td>0.71</td>
<td>0.4</td>
<td>0.8</td>
<td>1.0</td>
</tr>
<tr>
<td>T4</td>
<td>1.0</td>
<td>0.6</td>
<td>0.4</td>
<td>1.0</td>
<td>0.6</td>
<td>0.4</td>
<td>1.0</td>
<td>0.2</td>
<td>0.6</td>
<td>1.0</td>
<td>0.2</td>
<td>0.6</td>
<td>1.0</td>
<td>0.6</td>
<td>0.4</td>
</tr>
<tr>
<td>T5</td>
<td>0.6</td>
<td>1.0</td>
<td>0.4</td>
<td>0.6</td>
<td>1.0</td>
<td>0.8</td>
<td>0.6</td>
<td>0.2</td>
<td>0.4</td>
<td>0.2</td>
<td>0.4</td>
<td>0.2</td>
<td>0.4</td>
<td>0.6</td>
<td>1.0</td>
</tr>
<tr>
<td>T6</td>
<td>0.4</td>
<td>0.8</td>
<td>1.0</td>
<td>0.4</td>
<td>0.8</td>
<td>1.0</td>
<td>0.4</td>
<td>0.2</td>
<td>0.71</td>
<td>0.4</td>
<td>0.2</td>
<td>0.71</td>
<td>0.4</td>
<td>0.8</td>
<td>1.0</td>
</tr>
<tr>
<td>T7</td>
<td>1.0</td>
<td>0.6</td>
<td>0.4</td>
<td>1.0</td>
<td>0.6</td>
<td>0.4</td>
<td>1.0</td>
<td>0.2</td>
<td>0.6</td>
<td>1.0</td>
<td>0.2</td>
<td>0.6</td>
<td>1.0</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>T8</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>1.0</td>
<td>0.2</td>
<td>0.2</td>
<td>1.0</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>T9</td>
<td>0.6</td>
<td>0.4</td>
<td>0.71</td>
<td>0.6</td>
<td>0.6</td>
<td>0.71</td>
<td>0.6</td>
<td>0.2</td>
<td>1.0</td>
<td>0.6</td>
<td>0.2</td>
<td>1.0</td>
<td>0.6</td>
<td>0.6</td>
<td>0.71</td>
</tr>
<tr>
<td>T10</td>
<td>1.0</td>
<td>0.4</td>
<td>0.4</td>
<td>1.0</td>
<td>0.4</td>
<td>0.4</td>
<td>1.0</td>
<td>0.2</td>
<td>0.6</td>
<td>1.0</td>
<td>0.2</td>
<td>0.6</td>
<td>1.0</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>T11</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>1.0</td>
<td>0.2</td>
<td>0.2</td>
<td>1.0</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>T12</td>
<td>0.6</td>
<td>0.4</td>
<td>0.71</td>
<td>0.6</td>
<td>0.4</td>
<td>0.71</td>
<td>0.6</td>
<td>0.2</td>
<td>1.0</td>
<td>0.6</td>
<td>0.2</td>
<td>1.0</td>
<td>0.6</td>
<td>0.4</td>
<td>0.71</td>
</tr>
<tr>
<td>T13</td>
<td>1.0</td>
<td>0.6</td>
<td>0.4</td>
<td>1.0</td>
<td>0.6</td>
<td>0.4</td>
<td>1.0</td>
<td>0.2</td>
<td>0.6</td>
<td>1.0</td>
<td>0.2</td>
<td>0.6</td>
<td>1.0</td>
<td>0.6</td>
<td>0.4</td>
</tr>
<tr>
<td>T14</td>
<td>0.6</td>
<td>1.0</td>
<td>0.8</td>
<td>0.6</td>
<td>1.0</td>
<td>0.8</td>
<td>0.4</td>
<td>0.2</td>
<td>0.6</td>
<td>0.4</td>
<td>0.2</td>
<td>0.4</td>
<td>0.6</td>
<td>1.0</td>
<td>0.4</td>
</tr>
<tr>
<td>T15</td>
<td>0.4</td>
<td>0.4</td>
<td>1.0</td>
<td>0.4</td>
<td>0.4</td>
<td>1.0</td>
<td>0.4</td>
<td>0.2</td>
<td>0.71</td>
<td>0.4</td>
<td>0.2</td>
<td>0.71</td>
<td>0.4</td>
<td>0.4</td>
<td>1.0</td>
</tr>
</tbody>
</table>

As was previously noted in Section 3.3 the diagonal of the matrix shows the complete matches resulting from the comparison of a trajectory to itself. Additionally, the similarity values resulting from the LCS comparisons will be symmetric about the diagonal because similarity comparisons are not order dependent. In essence, a comparison between trajectories 4 and 6 will have the same similarity result as a comparison between trajectories 6 and 4. An analysis of a few selected attack tracks is broken down below with the aid of visual representation, using the severity attribute.

The example depicted in Figure 4.1 shows a visual representation of the severity attribute’s progression over time. This example reveals the validity and use of the LCS algorithm to detect similarities in attacks that are stretched out in time. Examination of Table 4.2 shows that the comparison of trajectory 3 and trajectory 15 leads to a complete match. This is because the two attacks follow the same severity progression over two different periods of time.

The example depicted in Figure 4.2 shows how the time windowing functionality was
used to detect similarities in trajectories that may not be immediately obvious upon visual inspection. In this instance, a 60 percent similarity match is found between the two trajectories as can be confirmed by examining Table 4.2. By simply realigning the example presented in Figure 4.2 to look more like the trajectories presented in Figure 4.3 this similarity can be more easily seen.

Using the trajectories in Figure 4.3 the resulting similarity value in Table 4.2 can be explained. Starting at the point on trajectory 4 that is at approximately time 14, this point can be matched to the point just above it on trajectory 2 that also occurs at approximately time 14. Then the next two points to the right on both trajectories will also be successfully matched. Through the use of the time windowing system, the point that is on trajectory 2 and occurs at approximately time 22 will be ignored as if it were noise. Then the final two points on trajectories 2 and 4 will be matched, creating 3 positive matches over the course of a minimum of 5 possibilities. Using the similarity measure from Equation (2.12) the final similarity value will thus be calculated as 0.6.
Figure 4.2: Two trajectories that illustrate a more complex similarity.

The last example, presented in Figure 4.4 illustrates two trajectories which have very few similar steps. Thus, the resulting similarity value as shown in Table 4.2 is a very low value of 0.2. Throughout all possible alignments of the two trajectories, no where do any sequences greater than a single point match occur. Furthermore, trajectory 8 is composed of 5 individual alerts, two of which occur very closely at approximately 8 seconds. Using Equation (2.12) leads to a value of 0.2 for the two trajectories presented here.

Although the severity attribute is the only one presented here, the concepts that are illustrated by these examples hold true for each of the other alert attributes examined.

4.3 The Relationship of Entropy Based on Trajectory Similarity

Recalling the similarity matrix shown in Table 4.2 the meaning of trajectory similarity was explored further in regards to the role it plays in determining importance. Without concrete examples and critical thinking simply applying the entropy value proposed by Dash et al. gives only minor insight to reasons behind the approach taken in this thesis.
Figure 4.3: Two trajectories, one of which has been realigned to more clearly show similarity.

An examination was performed with the focus of determining how the importance measure or entropy value corresponded to the similarities contained within an attribute’s similarity matrix. Furthermore, the resulting importance values were related back to the notion of cyber trajectory data.

It can be said quite trivially that a similarity matrix containing only the value 0.9 would result in an attribute that is not interesting. In terms of cyber trajectories this indicates a case where all the trajectories of a certain attribute can be classified into one group or as one type of cyber trajectory. Since there is little that can be learned from a data set containing trajectories that all belong to the same group, it can be said that an attribute with this characteristic would be of little importance. A data set consisting of all 0.9 similarity values would result in a corresponding entropy value of 0.4690.

In the opposite case, where a similarity matrix containing only values of 0.1 is considered, the result will be the same. Because no classifications or groping of trajectories can be performed, based on the fact that there is no similarity between any of the trajectories
Figure 4.4: Two trajectories that have a low similarity with each other.

present, then little knowledge can be gained from examining this data set. Therefore, the attribute that this type of similarity matrix represents shows little importance. A data set consisting of similarity values that were all 0.1 would result in an entropy value of 0.4690.

In a more complex manner, if a similarity matrix contains both the values 0.1 and 0.9 the result can still be used to classify an attribute as not important. This may seem contrary to intuition because values of both 0.1 and 0.9 represent some type of variation in a data set and therefore it may be desired to consider this interesting. However, consider a data set that contains 10 trajectories in which Trajectories 1,2,3,4, and 5 are all similar to each other with a value of 0.9 and the Trajectories 6,7,8,9, and 10 are all similar to each other with a value of 0.9. Additionally, consider a situation in which the trajectories from the first set are all similar to the trajectories from the second set with a value of 0.1 or %10 percent similarity. In this case it would be said that all the trajectories that represent this attribute can be strictly classified into one of two types. This by itself is not largely interesting.

On the contrary, in a case where data akin to that presented in Table 4.2 exists, there is
a considerable variability that can be used to determine the level at which this attribute is interesting. Since this data set contains trajectories with varying similarity values as shown in the histogram in Figure 4.5, it can be seen that many different classifications can be created.

![Histogram of Similarity Values](image)

*Figure 4.5: A histogram showing the varying values of similarity in Table 4.2.*

Examining Table 4.2 it can be said that since trajectory 14 has a %60 similarity with trajectory 1 then these two trajectories could be classified as the same type of attack and since trajectory 1 also has a %60 similarity with trajectory 12 the they could be included in this same classification as well. However, it can also be seen that trajectories 12 and 14 have a similarity with each other of only %40 percent so maybe they shouldn’t be in the same classification as each other. In this manner, many different groupings or classification schemes can be validly created thus making the attribute represented by this corresponding similarity matrix of great interest. The similarity matrix in Table 4.2 results in an entropy value of 0.7736.
These above cases give better overall intuition on why and under what context the entropy measure proposed by Dash et al. is being applied to cyber trajectory data.

## 4.4 Granularity Analysis of Attributes

Each of the attributes used during this thesis was examined based on two different levels of division. The resolution is the first type of division level examined. The resolution of an attribute can be defined as the number of inherent separate categories that an attribute can be divided up into. In the case of the severity attribute, this is limited to the extended list of different categories presented in Table 3.2 and based on Argauer’s original categories presented in [2]. The maximal resolution possible for the destination IP address attribute is to compare attributes simply by specific addresses. In this thesis however, the destination IP address will be analyzed at the subnet level. This is further explained in Section 3.1.2 and illustrated in Figure 3.1. The attribute services can have an extremely high resolution since there is a multitude of network services offered for many systems. Furthermore, each service attribute can be broken down into the different versions of that service that may exist. For the purposes of this thesis the resolution presented in Table 3.3 was used. Lastly, the protocol attribute is limited to the few network protocols that exist and are commonly used. The inherent resolutions that were used for each of the attributes in this thesis were discussed previously in Section 1.2.3.

The second level of division that exists for each of the attributes in this thesis is that of the granularity factor, $\alpha$ presented in Equation (3.1). This factor of granularity is different from the inherent resolution of each attribute in that it does not necessarily conform to the
inherent groupings or levels that exist for each of the attributes. For example, setting the granularity factor for the destination IP address attribute to a value of 10 is not the same just dividing the destination IP address attribute up into its different inherent subnets. This is constant for each of the different attributes presented except possibly the protocol attribute which has no further groupings or divisions and is categorical by nature.

When selecting the optimal granularity value for each attribute, it is important to use a value which is neither too big nor too small. If the granularity is too small singular comparisons will result in the set of matching points being too restricted and thus trajectories with good similarities may not be matched. If the granularity value is too high then singular comparisons will too easily result in a match and the resulting set of matching points will be too inclusive. In this thesis the optimal granularity factor is found by analyzing multiple random attack scenarios and then using the resulting entropies to determine which granularity values are the best. The entropy value here was determined using the previously specified entropy Equation (3.1).

Using the network simulation tool developed by Costantini, a large number of random attack trajectories was created. The LCS method was then used to obtain similarity matrices for each attribute which, were in turn used to compute the overall entropy of each attribute. The results are presented below in Table 4.3.

### 4.4.1 Alert Severity

The results in Table 4.3 show that the alert severity attribute has the highest entropy value at a granularity of 2. Furthermore, the high alert severity entropy value of 0.8519 indicates that this attribute does contain important trajectory information. It should also be noted that
### Table 4.3: Results of changing the granularity over a random assortment of attacks.

<table>
<thead>
<tr>
<th>Granularity</th>
<th>Severity</th>
<th>Destination IP Address</th>
<th>Services</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.7874</td>
<td>0.7156</td>
<td>0.8206</td>
<td>0.5936</td>
</tr>
<tr>
<td>1</td>
<td>0.7874</td>
<td>0.5691</td>
<td>0.8674</td>
<td>0.4186</td>
</tr>
<tr>
<td>2</td>
<td>0.8519</td>
<td>0.4388</td>
<td>0.8726</td>
<td>0.2979</td>
</tr>
<tr>
<td>3</td>
<td>0.8519</td>
<td>0.3673</td>
<td>0.8766</td>
<td>0.2979</td>
</tr>
<tr>
<td>4</td>
<td>0.8457</td>
<td>0.3164</td>
<td>0.8789</td>
<td>0.2979</td>
</tr>
<tr>
<td>5</td>
<td>0.8457</td>
<td>0.2979</td>
<td>0.8783</td>
<td>0.2979</td>
</tr>
<tr>
<td>10</td>
<td>0.6814</td>
<td>0.2979</td>
<td>0.8180</td>
<td>0.2979</td>
</tr>
<tr>
<td>25</td>
<td>0.3502</td>
<td>0.2979</td>
<td>0.6618</td>
<td>0.2979</td>
</tr>
</tbody>
</table>

Unlike the destination IP address attribute, the severity attribute does not have to follow a specific path to any end goal. As a result, the alert severity trajectory may contain many variations of sequences of the exposure damage scores discussed in 3.1.1. Recall that these values were developed as an expansion to Argauer’s approach to assigning damage scores to certain exposures in [2]. Moreover, these score values provide the basic resolution that is used by the alert severity attribute throughout this thesis.

### 4.4.2 Destination IP Address/Subnet

The destination IP address, which is examined at the subnet level, has a maximal entropy at a granularity of 0. Since attacks must follow a specific path on their way to many places in the network, it is highly likely that there is a significant number of trajectories with high subsequence similarities for this attribute. For example, an attack on Zone 4 of the Network from Figure 3.1 would most likely be a subsequence of an attack on Zone 7. This is because both attacks will have to go through the progression of attacking Zone 1, then attacking Zone 2, and then attacking Zone 4. This idea in conjunction with the fact that the maximum entropy value occurs at a granularity of 0 may suggest that the Destination
IP address attribute should be examined at a lower resolution. However, the next lowest inherent resolution would occur at the individual address level. This would easily create an attribute which is too divided and would thus be too restrictive during comparisons. Additionally, using the granularity factor to compensate for this increased division would be inappropriate as it would too easily allow for matches between alerts that occur in two different subnets. Moreover, the resulting maximum entropy value presented in Table 4.3 is significantly high enough to say that this attribute is important at this level of resolution.

4.4.3 Services Attacked

The services attribute has an entropy value near the maximum for granularities between 1 and 5 with an overall maximum entropy occurring at a granularity of 4. This indicates that the selected resolution for the services attribute provides an appropriate level of inherent division. Recall that the inherent resolution for this attribute was selected by grouping services with similar cyber properties and attack characteristics and is shown in Table 3.3 along with further discussion in Section 3.1.3.

In general, networks are setup to offer more than one service to a user, thus allowing for a multitude of different overall paths for the service trajectory to take. However, in some cases certain machines or groups of machines may be setup to offer only one or a limited number of services. In this instance, the reduced set of possible services that can be attacked essentially creates one path which a hacker must follow to obtain certain end goals on a defined portion of a multi-level network. As a result, the services attribute would no longer become useful in this type of network configuration because all of the attacks would have to follow that service path and thus a significantly high number of attacks would be
complete matches.

4.4.4 Protocol Used

The protocol attribute suffers from the fact that there are only so many divisions which are made thus giving a very restricted overall granularity. Due to the inherent nature of the protocol attribute, it cannot be divided up into a finer grained resolution for examination. This in general contributes to conclusion that the protocol attribute is of very little use for this type of trajectory analysis.

4.5 Network Topology

As was discussed above, the importance of some alert attributes may be altered by changes in an overall network configuration. One place in which this is highly notable is in the setup of a network’s topology. In particular, it is expected that changes in the network topology could have an adverse effect on the destination IP address/subnet attribute. An examination of two network extremes, a network with a large breadth and a network with a large depth, was performed. This breadth versus depth characteristic was examined using the two networks presented in Figures 4.6 and 4.7. The two networks were setup to be the same in all characteristics except for the network topology. Then the same attack vectors were ran on each network over a wide range of granularities. The resulting entropies are recorded in Table 4.4 and Table 4.5.

Examination of these results quickly shows that the subnet attribute suffers in the increased depth network topology. This is reflected in the resulting low subnet attribute entropy values for all the possible granularities shown in Table 4.5. This is a direct result
Table 4.4: Random attacks over the increased breadth network.

<table>
<thead>
<tr>
<th>Granularity</th>
<th>Severity</th>
<th>Destination IP Address</th>
<th>Services</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.7958</td>
<td>0.8706</td>
<td>0.8911</td>
<td>0.5439</td>
</tr>
<tr>
<td>1</td>
<td>0.7958</td>
<td>0.7753</td>
<td>0.8843</td>
<td>0.5439</td>
</tr>
<tr>
<td>2</td>
<td>0.8434</td>
<td>0.6528</td>
<td>0.8843</td>
<td>0.4690</td>
</tr>
<tr>
<td>4</td>
<td>0.8570</td>
<td>0.5303</td>
<td>0.8843</td>
<td>0.4690</td>
</tr>
<tr>
<td>10</td>
<td>0.7277</td>
<td>0.4690</td>
<td>0.8434</td>
<td>0.4690</td>
</tr>
<tr>
<td>25</td>
<td>0.4894</td>
<td>0.4690</td>
<td>0.8162</td>
<td>0.4690</td>
</tr>
</tbody>
</table>

Table 4.5: Random attacks over the increased depth network.

<table>
<thead>
<tr>
<th>Granularity</th>
<th>Severity</th>
<th>Destination IP Address</th>
<th>Services</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.7553</td>
<td>0.3568</td>
<td>0.6666</td>
<td>0.3531</td>
</tr>
<tr>
<td>1</td>
<td>0.7553</td>
<td>0.2754</td>
<td>0.6511</td>
<td>0.3531</td>
</tr>
<tr>
<td>2</td>
<td>0.7727</td>
<td>0.2754</td>
<td>0.6288</td>
<td>0.2754</td>
</tr>
<tr>
<td>4</td>
<td>0.7580</td>
<td>0.2754</td>
<td>0.6288</td>
<td>0.2754</td>
</tr>
<tr>
<td>10</td>
<td>0.7301</td>
<td>0.2754</td>
<td>0.6016</td>
<td>0.2754</td>
</tr>
<tr>
<td>25</td>
<td>0.3681</td>
<td>0.2754</td>
<td>0.4151</td>
<td>0.2754</td>
</tr>
</tbody>
</table>

of the ease with which subnet trajectories can have matching sequences. Since every attack on the increased depth network must move through the same subnets on their way to a final goal, there will be a significantly high number of very similar subsequences. For this reason, the subnet attribute becomes much less interesting and this is reflected in the reduced entropy values that result.

Although similar results would be expected on the increased breadth network, where it would be expected that very few trajectories followed the same path, the actual results given in Table 4.4 show that an acceptable entropy value is obtained at a granularity of zero. This result can be explained by the fact that a basic principle of the layered network approach presented by Argauer is still maintained [2]. Using a set of servers connected to the internet to help with control of network traffic forces attacks to go through this
subnet and therefore creates at least some similarity in the resulting attack trajectories. This means that some similarities in the subnet attribute will be inherent in the resulting data and therefore, acceptable entropies will result.

![Network with an increased breadth](image)

**Figure 4.6: Network with an increased breadth**

On the contrary, the severity attribute should constantly be a useful attribute for the characterization of a hackers movement on a network. This is because the severity attribute is not as easily effected by changes in the network topology as the subnet or services attributes. Over the many different attack scenarios that were run, the severity attribute always had a value that was significantly high enough to distinguish it as an important attribute as can be seen in Tables 4.4 and 4.5.

### 4.6 Efficiency and Stealth Factor

The network simulation tool developed by Costantini in [4] includes adjustable parameters that are used to mimic the different behavioral patterns that hackers may exhibit. The
parameters stealth and efficiency were varied during this thesis to create realistic attack scenarios. The stealth parameter is used to set how directly a hacker can reach his end goal [4]. A low stealth value (close or equal to 0.0) shows many or all of the intermediate steps that a hacker has taken in order to reach a final goal [4]. A high stealth value (close or equal to 1.0) represents an attack in which only a few intermediate steps or only the final goal step is reported by the IDS [4]. Similarly, the efficiency parameter varies how directly a resulting attack moves towards an end goal. An attack that is generated with an efficiency value of 0.0 will choose many intermediate steps, including many steps that could be misleading from an analysts standpoint [4]. This can be pictured as an attack that
branches out in many directions before reaching its final goal [4]. In contrast, an attack with an efficiency factor of 1.0 will result in an attack that has very few or no branches and each step represents a direct approach to the overall goal [4].

In order to further examine the effects of increased noise in attack scenarios and to examine the merits of the implemented time windowing system, multiple random low efficiency and high stealth attacks were ran on a network. The resulting attack trajectories generated should contain both a significant amount of noise as well as many unreported steps. This will test the algorithm’s ability to handle noise and missing alerts. Additionally, many different attack targets were chosen in order to create tracks of various lengths. The resulting entropy value over a range of granularities is given in Table 4.6.

<table>
<thead>
<tr>
<th>Granularity</th>
<th>Severity</th>
<th>Destination IP Address</th>
<th>Services</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.8392</td>
<td>0.8911</td>
<td>0.7816</td>
<td>0.5125</td>
</tr>
<tr>
<td>1</td>
<td>0.8392</td>
<td>0.8567</td>
<td>0.8086</td>
<td>0.2648</td>
</tr>
<tr>
<td>2</td>
<td>0.8662</td>
<td>0.5748</td>
<td>0.8433</td>
<td>0.1095</td>
</tr>
<tr>
<td>4</td>
<td>0.8926</td>
<td>0.2369</td>
<td>0.8701</td>
<td>0.1082</td>
</tr>
<tr>
<td>10</td>
<td>0.8049</td>
<td>0.1082</td>
<td>0.8708</td>
<td>0.1082</td>
</tr>
<tr>
<td>25</td>
<td>0.3290</td>
<td>0.1082</td>
<td>0.6131</td>
<td>0.1082</td>
</tr>
</tbody>
</table>

Table 4.6: Random attacks on a large network with low efficiency (0.1) and a high stealth (0.9).

The result of these attacks shows that the time windowing and time stretching detection capabilities do indeed work well in cases where noise and missing alert data is present. This shows that the usefulness of the time windowing system as well as the inherent ability to detect time stretched attacks goes well beyond just the simple verification capabilities. Furthermore, by examining the results it can be seen that the subnet attribute still provides significantly important information even though the low efficiency factor is generating many
noisy alerts. However, the increased amount of noise in the attack scenarios does appear to have a slight effect on the services attribute which, in this case, functions better at a higher granularity value than in most other cases. This result in comparison to the result of the subnet attribute are in high contrast due to the fact that the subnet attribute still has a defined path to follow and after the noise is handled by time windowing this path can still be found. In the case of the services attribute, the noisy attacks will most likely create inconsistencies in the histogram used to keep track of the popularity of attacks on a certain service. Therefore, the resulting service values match with a low similarity and must use a higher granularity to compensate for the increased difference. This was confirmed to be the case by examining the similarity matrices produced by both of these attributes. The entropy values for the protocol attribute, once again are significantly lower than for each of the other attributes studied.

4.7 Results

Using the granularity values specified above, the LCS method was run and then entropy values were determined for each of the attack scenario data sets created. The resulting data is presented in Table 4.7.

As can easily be seen from the entropy results in Table 4.7, the protocol attribute consistently provides a significantly lower value than the other three attributes in every attack scenario tested. This shows directly that the progression of the protocol attribute’s trajectory constantly contains the least amount of interesting information. Moreover, it was determined that the other three attributes, alert severity, destination IP address, and services
<table>
<thead>
<tr>
<th>Attacks</th>
<th>Severity Entropy</th>
<th>Destination IP Address Entropy</th>
<th>Services Entropy</th>
<th>Protocol Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1</td>
<td>0.8450</td>
<td>0.9148</td>
<td>0.9139</td>
<td>0.6459</td>
</tr>
<tr>
<td>Scenario 2</td>
<td>0.7218</td>
<td>0.8091</td>
<td>0.8392</td>
<td>0.5425</td>
</tr>
<tr>
<td>Scenario 3</td>
<td>0.8306</td>
<td>0.7010</td>
<td>0.8283</td>
<td>0.5933</td>
</tr>
<tr>
<td>Scenario 4</td>
<td>0.8475</td>
<td>0.7068</td>
<td>0.8575</td>
<td>0.5574</td>
</tr>
<tr>
<td>Scenario 5</td>
<td>0.8538</td>
<td>0.7187</td>
<td>0.8726</td>
<td>0.5353</td>
</tr>
<tr>
<td>Scenario 6</td>
<td>0.8241</td>
<td>0.7305</td>
<td>0.8513</td>
<td>0.5541</td>
</tr>
<tr>
<td>Scenario 7</td>
<td>0.6592</td>
<td>0.6650</td>
<td>0.8452</td>
<td>0.3509</td>
</tr>
<tr>
<td>Scenario 8</td>
<td>0.8026</td>
<td>0.7238</td>
<td>0.8541</td>
<td>0.6023</td>
</tr>
</tbody>
</table>

Table 4.7: Entropy results over multiple attack scenarios

attacked, provide meaningful and interesting data over all attack scenarios tested.

In the more extreme cases testing the effects of network topology and attack efficiency revealed distinct characteristics about each attribute. Although the destination IP address attribute performs well in most case, it does however suffer in a network where an increased depth is present. Moreover, it can be said that the destination IP address attribute is important and contains useful trajectory information as long as the network has an evenly distributed network depth and breadth. Similarly, the services attribute can suffer when the only a few or a single service is made available on network machines. This restricted path will cause an artificially large number of positive matches to occur and will thus make the services attribute less interesting. This becomes somewhat apparent in the increased depth network where it is highly likely that a large number of the same service paths are used. The decrease in the importance of this attribute is reflected in the lower entropy value found in Table 4.5. The severity attribute performs well over all attacks and is therefore not only the most interesting attribute but also the most robust of the attributes presented. Lastly, in approximately all the cases presented, the protocol attribute consistently performed poorly.
For this reason, it was determined that the protocol attribute is of little interest in terms of examining cyber attack tracks in a trajectory based fashion.
Chapter 5

Conclusions

Overall, this work was able to apply a subsequence matching method to cyber data in order to discover similarities within attack trajectories. Furthermore, a measure of entropy was used to quantify the importance of each of the attributes being examined. The combination of these two techniques allowed trajectory and sequence information to be taken into account when performing dimensionality reduction. Additionally, qualitative and quantitative analysis methods were used to determine the appropriate levels of division or resolution and granularity for each of the attributes being considered. Using the network simulation tool developed by Costantini, many other factors including network topology and attack efficiency were thoroughly tested and analyzed.

This work has discovered and confirmed that the protocol attribute is the least important and least meaningful of the alert attributes examined. Furthermore, analysis of multiple attack scenario data sets has confirmed that the other three alert attributes examined, alert severity, services attacked, and destination subnet attacked, are of significant importance in terms of the progression of data contained within their corresponding trajectories. Moreover, it was discovered that a measure of entropy can be applied to the similarity matrix resulting from an appropriate sequence comparison method to discover the most important
alert attributes. In conclusion this work can accurately apply a dimensionality reduction technique while still accounting for the order progression of an attack in the form of a trajectory.

5.1 Future Work

In the future this method could be used with a system that has a large number of reliable alert attributes for the reduction of data dimensionality in order to make cyber alert data easier to examine. Furthermore, this method could easily be applied to other problem domains in which trajectories or ordered data sequences of high dimensionality must be considered. In particular, this method could be applied to the asymmetric warfare domain to discover which strategic events are of the greatest importance.

Much future work can also be done to examine different resolutions of each attribute but, will require a more powerful, robust, and inclusive simulation tool. Furthermore, a larger number of alert attributes could be examined given that they are reliable enough. The analysis of actual cyber alert data from existing networks could also be used to aid in increasing the validity of results data.

Use of this work has already been applied to the hierarchical clustering method implemented in thesis work done by Murphy [25]. The work done by Murphy seeks to use social networking and hierarchical clustering concepts to perform clustering on network services attacked over time. The LCS method from this thesis is being used to develop similarity matrices for each of the alert attributes being examined. The hierarchical clustering methods used by Murphy are the being applied to these matrices to obtain a better overall
characterization of a cyber hacker’s actions on a network. Continual research is being performed to investigate the merits of running this combination of algorithms in real-time for the purpose of characterizing a cyber hacker’s capability. Additionally, this combination of work could lead to a system which is able to provide a sufficiently accurate method of predicting future alerts and cyber hacker actions.
Bibliography


