Data Mining Techniques To Analyze Network Intrusion Pattern

Christopher Cowden Michael Samidi

Department of Computer Science
University of California
Riverside, CA 92521
{ccowden, msamidi}@cs.ucr.edu

Abstract

In this project, we analyze data mining techniques that are used to detect network intrusion patterns. Intrusion detection has been studied for awhile but recently more attention has been given to defending network resources as opposed to host systems from intrusions. For this project, we focus primarily on TCP protocol since this protocol is very widely used today. Using the network *TCPDump* data from the Shootout Project, we will evaluate standard feature selection techniques by comparing the classifiers that are generated when using these techniques.

1 Introduction

Network intrusion is becoming a major issue as more and more end hosts are added to the Internet. As we can see from the growth of the Internet, we will face even more challenging tasks to secure our computer system from any kind of intrusions in the near future. There are some known mechanisms to prevent these attacks, particularly through passive or active network analysis. However, for this project we concentrate more on analyzing *TCPDump* data collected from the Information Exploration Shootout Project [8]. Particularly, we are interested in using data mining techniques to analyze the data.

The data consists of a baseline file (normal traffic) and four intrusion files. We construct our classification rules from the baseline and apply the rules into the intrusion files. Because the combined datasets are extremely huge (~200 MB), we use SGI MineSet tools to automate our analysis step. The software has a rich set of features such as classification, regression and clustering for data mining analysis.

The rest of the paper is organized as follows. In Section 2, we present some background of network intrusion, TCP/IP protocol suite and data mining techniques. In Section 3, we present the experiment results based on the *TCPDump* data. Finally, we conclude in Section 4 and outline an extension for this project in Section 5.

2 Background

In the following section, we try to present some basic knowledge about network intrusion. To better understand about our problem, we pose five fundamental questions and try to answer them the best we can. Furthermore, we will also discuss briefly about TCP/IP protocol suite. Finally, we give some data mining techniques from SGI MineSet tool that we use to analyze the data.

2.1 Network Intrusion

We will provide some basic information for network intrusion that is relevant to this project.

2.1.1 What is a network intrusion?

Basically, a network intrusion is "any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource" [7].

2.1.2 How do network intrusions happen?

Most of the time, network intrusions happen because there are some security flaws in the network due to design and programming errors. For example, TCP sequence number prediction, denial of service, the "authentication" server and so on [6].

2.1.3 Who performs a network intrusion?

Note that we do not make a distinction between "attackers" and "intruders". We think this distinction is best left to language lawyers and the key point to note is that all unauthorized use of network resources should be categorized together. To answer our question, most of the attackers are "insiders" that exploit any weaknesses of the network for fun or profit although external deviants certainly contribute to the problem. For example, imagine a scenario where the first party (you) and the second party (the owner of a host who connects to your network) make a crucial conversation on the network. Furthermore, neither you nor the second party can identify a third party (the "truly guilty" adversary) who is performing a man-in-the-middle attack by intercepting the first host's communications and impersonating the first in their communication to the second host. In this attack, not only is the attacker gaining valuable information but, the attacker is also utilizing the first host's identity (and possibly their network resources) to attack the second party (the "true" victim).

2.1.4 When does network intrusion happen?

The attacks happen all the time. To our knowledge, there has not been a significant analysis of when attacks occur although we will guess that they vary with the types of attacks as well as the types of attackers perpetrating the offense.

2.1.5 Where does network intrusion take place?

Usually, it happens where the network is weakest. This is called the principle of least effort and it means what it implies; namely, that the attacker will go to the least trouble possible to get what they want. We will study attacks on the TCP protocol although we don't mean to imply anything regarding the difficulty of attacking the TCP protocol.

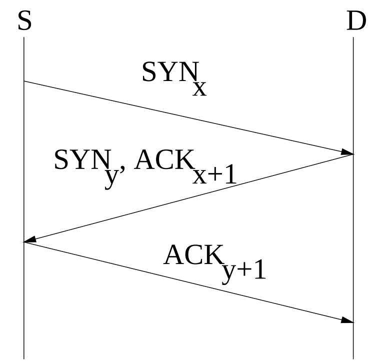

Figure 1: Three-way Handshake

2.2 TCP/IP Protocol Suite

We will provide a brief discussion about the TCP/IP protocol. We studied the protocol intensively, especially its security related problems [3, 4, 5, 6, 9, 11, 12]. Here we give a more detailed overview about the protocol and the important features that characterize a connection.

2.2.1 Internet Protocol (IP)

The *Internet Protocol* (IP) is the standard network layer protocol of the Internet that provides an unreliable, connection-less, best-effort packet delivery service. IP provides the basic unit of data called the *datagram* from a source to a destination, where sources and destinations are hosts identified by fixed length addresses. The service is *unreliable* because the delivery of the datagram is not guaranteed. IP is *connection-less* because each packet can travel over different paths.

2.2.2 Transmission Control Protocol (TCP)

To ensure reliable communication for applications and services, the *Transmission Control Protocol* is provided. TCP provides a *reliable*, *connection-oriented* data stream delivery service. Before data can be transmitted between the source host S_i and destination host D, TCP needs to establish a connection between S_i and D. This connection setup is called the *three-way handshake* (see Figure 1). We illustrate only the normal connection setup. First, the source S_i sends the SYN packet with a sequence number, x to the destination D. Then, D sends back to S_i both the SYN packet with a sequence number, y and ACK packet with a sequence number, $x + 1$. This is to indicate that D acknowledges the SYN packet and is continuing the handshake. Finally, S_i sends ACK with a sequence number $y + 1$ and is an indication to D that both hosts agree that a connection has been established. The third packet may contain user payload data. Therefore, sequence numbers are very important indication of network intrusion. The flags such as SYN, ACK and so on are also an indication for any attacks.

2.3 Data Mining Techniques

In this section, we give some overview of the SGI MineSet data mining techniques that we use for this project.

Feature Selection: This is the most important step for our project. This step is to determine the relations between fields in the database. Analyzing the correlations in audit data provides insight in how to select the correct set of attributes for the data classification step. MineSet tools has a feature called *Column Importance* which determines the importance of specific columns in discriminating one label value from another. This feature is used to observe the varying effects of changing variables, or to suggest columns to map to the axes of the Scatter and Splat Visualizers.

Data Clustering: This is the important step in our analysis. The clustering algorithm segments the data into groups of records, or clusters, that have similar characteristics. The MineSet suite of tools has a *Clustering Algorithm* that groups data according to similarity of characteristics, then displays it as a series of box plots and histograms. The clustering algorithm displays results using the Cluster Visualizer.

Visual Data Mining *"An analytical data mining can be complemented with data visualization techniques taking advantage of the human brain's amazing pattern recognition capability"* [1]. For our project, we use the Scatter Visualizer extensively. The Scatter Visualizerworks as follows: data points are shown in one, two, or three dimensions. Additional attributes can be mapped to color, size, and shape. Finally, two additional attributes may be mapped to sliders, allowing animation and fly-throughs, for a total of eight dimensions. The column importance operation in MineSetcan help you identify the important dimensions to map for a given task and we use this as our method selecting the most relevant features.

3 Experiments on *TCPDump* Data

3.1 The *TCPDump* Data

A quick discussion of TCPDump output will explain most of mysterious traffic in the data. TCP-Dump datasets are available at "iris.cs.uml:8080/network.html", that is part of an Information Exploration Shootout [8]. The data set consists of 4 runs of TCPDump on generated network intrusions and one TCPDump run on normal network traffic. The output of each TCPDump run is in a separate file.

The general format of TCPDump output is:

```
time src > dst: flags data-seqno ack window urgent options.
```

Time is in the form of hour:minutes:seconds. *Src* and *dst* are the source and destination IP addresses and ports in the form of *x.y.256.256.port*. *Flags* are some combination of **S**(**SYN**), **F**(**FIN**), **P**(**PUSH**), **R**(**RST**) or '.' (**no flags**). *Data-seqno* describes the portion of sequence space covered by the data in this packet. *Ack* is the sequence number of the next data expected the other direction on this connection. *Window* is the number of bytes of receive buffer space available the other direction on this connection. *Urg* indicates there is 'urgent' data in the packet. *Options* are TCP options enclosed in angle brackets (e.g., <mss 1024>). *Time*, *src*, *dst* and *flags* are always present. The other fields depend on the contents of the packet's TCP protocol header and are output only if appropriate.

Filename	Number of records
baseline	358760
network1	628775
network2	481851
network3	509263
network4	632036

Table 1: The datasets

Since TCPDump output is not intended specifically for security purposes, we had to go through multiple iterations of data pre-processing to extract meaningful features and measures.

3.2 Data Pre-processing

We wrote our own Perl script to scan TCPDump data file and extract the "connection" level information about the network traffic. We had to revise the script many times because we wanted to make sure we captured all the meaningful information. The above table summarizes the number of records on each file. For the brevity of our project, we were only able to analyze the normal dataset (baseline) and the first two network intrusion files (network1 and network2).

3.2.1 Scanning Process

The scanning process converts TCPDump output into the following fields:

- **time** - converted into floating point seconds (hr*3600+min*60+secs)
- **addr** and **port** - the first two fields of the src and dst address make up the fake address, so it is converted into $(x + y)/1000$
- **flag** - "U" for UDP packet and **S**(**SYN**), **F**(**FIN**), **P**(**PUSH**), **R**(**RST**) or '.' (**no flags**) for TCP packet
- **seq1** - the data sequence number of the packet
- **seq2** - the data sequence number of the data expected in return
- **buf** - the number of bytes of receive buffer space available
- **ack** - the sequence number of the next data expected from the other direction on this connection
- **win** - the number of bytes of receive buffer space available from the other direction on this connection
- **ulen** - if a UDP packet, then it is the length
- **op** - optional info such as (df) - do not fragment

3.2.2 Extraction Process

The extraction process summarizes the above packet information into the "connection" level information. For each connection, the script processes packets between the two ports of the participating hosts, and:

- checks whether 3-way handshake has been properly followed to establish the connection. From our observation, it seems that all connection from baseline and network intrusion datasets establish 3-way handshake.
- monitors each data packet and ACK packet, keeps a number of counters in order to calculate these statistics of the connection: duplicate ACK rate, wrong (data packet) size rate, (data) bytes sent in each direction, percentage of data packet, and percentage of control packet.
- watches how connection is terminated: normal (both sides properly send and receive FINs), abort (one host sends RST to terminate, and all data packets are properly ACKed), half closed (only one host sends FIN), and disconnected.

We treat each UDP packet as a connection. Finally, this process produces connection records for each dataset which are now ready to be analyze using SGI MineSetdata mining tools. For each connection record, it has the following fields (features):

```
start time, duration, participating hosts, ports, the
statistics of the connection, connection status, termination
status and protocol (TCP or UDP).
```

From the ports, we know whether the connection is to a well-known service, e.g., *http* (port 80), or a user application.

3.3 Experiments and Results

Step 1: Projection In order to generate informative clusters we first removed attributes that gave us no extra information; for example, those attributes for which every dataset had only 1 value. We removed the following attributes from our dataset: connection_status, resent_rate, wrong_resent_rate, dup_ack_rate, hole_rate, and wrong_size_rate.

Step 2: Filter Next, filters were added to help the SGI MineSettool select the appropriate attributes. The filters we applied were as follows: dst_port <= 5000, duration <= 1,000, and protocol == "TCP". The first two filters were decided upon by analyzing some preliminary gnuplots that we generated from the data as shown in Figures 2, 3, and 4. It is obvious to see that the majority of data points lie within the ranges implied by the filters save one grouping of points around the destination port value of 8000. After careful analysis, this grouping seems to be an outlier so it is not included.

Step 3: Cluster After narrowing down the range of selected attributes we pass the resulting dataset through the SGI MineSetcluster utility. The output of this cluster utility is a model of the dataset.

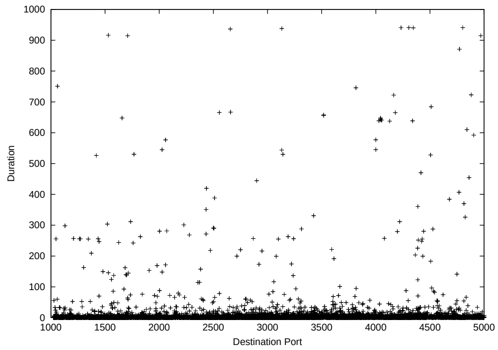

Figure 2: Destination Port vs Duration for Baseline file

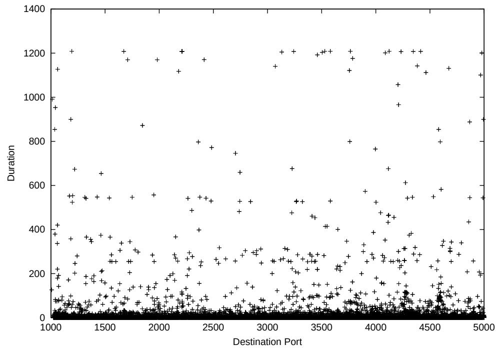

Figure 3: Destination Port vs Duration for Network1 file

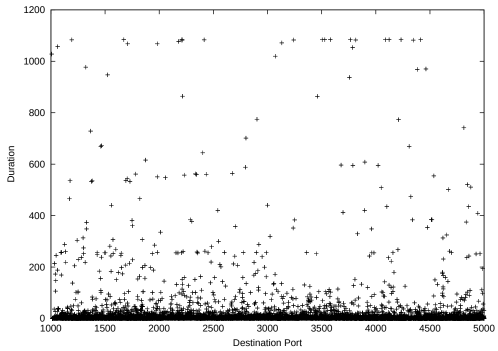

Figure 4: Destination Port vs Duration for Network2 file

Step 4: Apply Model We then apply the cluster model to the Scatter Visualizer utility provided with MineSet. This utility allows us to see how the attributes relate to each other between clusters as opposed to looking at them independently.

Step 5: Column Importance We then choose the three attributes that allow us to best view the clusters. This allows us to visually see the clusters based on the values of the data points in 3-dimensional space where each dimension is defined by the three chosen attributes. These attributes are chosen with another mining tool provided in MineSetthat is called Column Importance. We simply choose a label against which the importance is measured, and in our case we want the label to be the clusters generated during the clustering step. In other words, the three attributes will be chosen with respect to how important they to defining the clusters. We decided to generate the five most important attributes and then choose the best three among the baseline, network1, and network2 datasets in order to settle discrepancies between the attributes chosen. The results for each dataset is as follows:

Baseline:

- dst_port
- termination_status
- dst_ip
- percent_of_data_packet
- src_ip

Network1:

- dst_port
- termination_status
- dst_ip
- percent_of_data_packet
- src_ip

Network2:

- dst_port
- termination_status
- percent_of_data_packet
- dst_ip
- src_ip

The first two attributes are the same among all datasets and so these two are definitely included as the two most important descriptors of the clusters. The third was decided as follows: dst_ip can not be a good descriptor of a cluster since the values that the attribute can have are not ordered in any way. Thus, whatever meaning that the Column Importance utility is associating between the dst_ip attribute and the clusters found, is misguided and ill-founded. So we choose the next best attribute which is the percent_of_data_packet attribute.

Step 6: Plot Now we generate the scatter-plots for each dataset with Axis1 = dst_port, Axis2 = termination_status, Axis3 = percent_of_data_packet. We also set the color of each data point in this 3-dimensional plot to be represented by the cluster attribute so we can visually see the clusters by their color.

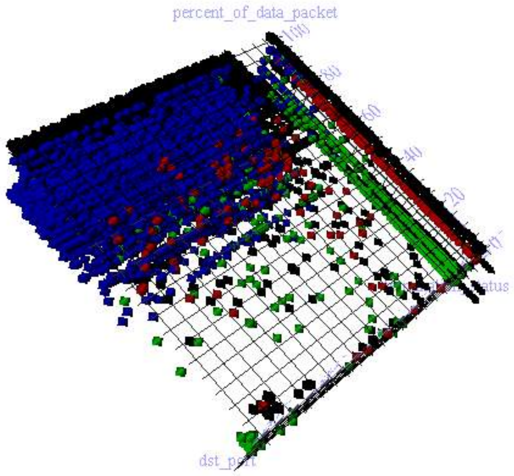

Figure 5: Baseline

Step 7: Analysis Notice the Figures 5, 6, and 7 which are generated from Clustering Visualizer. These Figures represent the clusters generated by MineSetfor each of the different datasets baseline, network1, and network2. Please note that the baseline data is our guideline in identifying an attack. After careful analysis, we assumed that since the largest difference between the two network plots and the baseline plot lies in the red cluster, that the red cluster would be the greatest indicator of an attack. In particular, the difference between the baseline and each of the network datasets appears to be unique. (Note: When we refer to the well known ports we mean those ports which are used for well known services such as http, ftp, finger, etc. The port numbers from 0 to 255 have been reserved for these ports.) Notice in the Network1 plot (Figure 6) how the data points composing the red and green clusters, whose dst_port value is a well known port, differ from the Baseline plot in Figure 5. This difference among the red and green clusters over the well known ports is immediately recognizable. Now Notice the Network2 plot in Figure 7. There is no noticeable difference among the clusters over the well known ports, when compared with the Baseline plot, but there is another difference that immediately jumps out. In the Network2 plot there seems to be a large grouping of red data points in the dst_port range of 1,250 to 5,000 and the percent_of_data_packet range of 50% to 75%. Although the Baseline plot has a similar grouping, it is not nearly as substantial as the Network2 red clustering in this range, and could indicate abnormal activity or an attack.

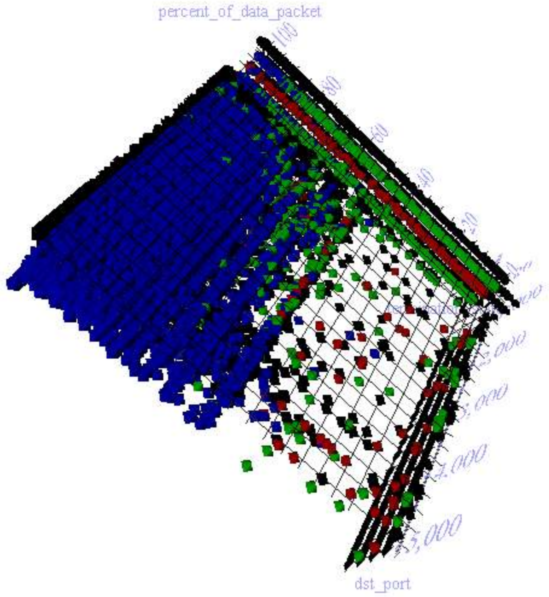

Figure 6: Network1

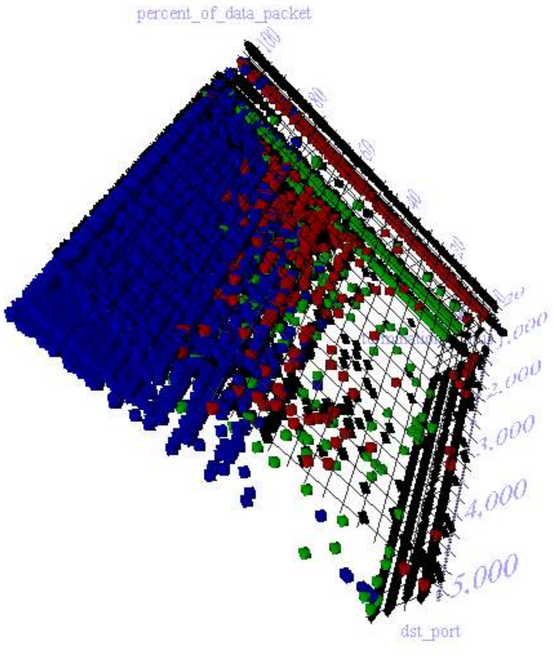

Figure 7: Network2

4 Conclusion

The purpose of our project is to identify any network intrusion using data mining techniques. Since most of the real-world traffic uses TCP/IP protocol, we focused on analyzing TCPDumpdata. The experiments on TCPDumpdata showed the effectiveness of clustering models in detecting any traffic anomalies. We found that the accuracy of the analysis definitely depends on the right set of features selected. Moreover, we found that the attacks are very difficult to detect. It requires security experts to analyze the behavior of the attacks. However, we found that SGIMineSettools helped us in automating any data mining techniques. It also has a rich set of data mining features that anyone can just use it off-the-shelf. We found that it is very easy to learn all the basic features of the tools.

5 Future Works

Our project can be extended in many ways. Our analysis is only the first step in building a complete model of intrusion detection. We need to find a consistent way to select meaningful attributes. This selection process definitely influences our determining malicious attacks. From literature searches, we found that most of the security papers do not have a suggestion on how to analyze TCPDumpdata using data mining techniques. Therefore, it is still an open-research issue on how to effectively using data mining techniques for the data. One way to solidify any results gained in this project and future work using data mining techniques is to have detailed domain knowledge of the specific attacks that are present in any datasets analyzed. This will give a direct feedback measure of the importance of any techniques utilized or methods used in analyzing the data. A different direction that could be taken, in the absence of this domain knowledge, would be to conduct an exhaustive and rigorous analysis of the data independent of any domain knowledge. Analyzing correlations between attributes in and between datasets, analyzing trends of attributes independently in and between datasets, as well as doing each of these at different levels of attention in the hopes of creating some kind of abstract concept hierarchy that could be utilized to further expertise in the current domain.

References

[1] Silicon Graphics, Inc., MineSet 3.1 Enterprise Edition Tutorial for Windows, "*http://www.sgi.com/software/mineset/demos.html*"

[2] Lawrence Berkeley National Laboratory, "*ftp://ftp.ee.lbl.gov/TCPDump.tar.Z*"

[3] J. Postel, editor, "*RFC-791 Internet Protocol*", Information Sciences Institute, USC, CA, Sept 1981

[4] J. Postel, editor, "*RFC-793 Transmission Datagram Protocol*", Information Sciences Institute, USC, CA, Sept 1981

[5] W. R. Stevens, "*TCP/IP Illustrated, Volume 1*", Addison-Wesley Publishing Company, 1984

[6] S. M. Bellovin, "*Security Problems in the TCP/IP Protocol Suite*", Computer Communication Review, Vol 19, No. 2, pp. 32-48, April 1989

[7] R. Heady, G. Luger, A. Maccabe, and M. Servilla, "*The Architecture of a Network Level Intrusion Detection System*", Technical Report, Computer Science Department, University of New Mexico, August 1990

[8] Information Exploration Shootout Project, "*http://iris.cs.uml.edu:8080*, 1997

[9] V. Paxson, "*End-to-end Internet Packet Dynamics*", In Proc. of SIGCOMM'97, September 1997

[10] W. Lee and S. J. Stolfo, "*Data Mining Approaches for Intrusion Detection*", In Proc. of the 7th USENIX Security Symposium, San Antonio, TX, 1998

[11] V. Paxson, *Bro: A System for Detecting Network Intruders in Real-Time*", In Proc. of the 7th USENIX Security Symposium, San Antonio, TX, 1998

[12] P. A. Porras and A. Valdes, "*Live Traffic Analysis of TCP/IP Gateways*", In Proc. of the Internet Society Symposium on Network and Distributed System Security, March 1998

[13] J. Han and M. Kamber, "*Data Mining: Concepts and Techniques*", Morgan Kaufmann Publishing Company, 2001