Case Study: Visualization and Information Retrieval Techniques for Network Intrusion Detection

Travis Atkison, Kathleen Pensy, Charles Nicholas,
David Ebert, Rebekah Atkison, Chris Morris

Computer Science and Electrical Engineering Department
University of Maryland, Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250
{atkison, kpensy1, nicholas, ebert, ratkis1, cmorris}@umbc.edu

Abstract. We describe our efforts to analyze network intrusion detection data using information retrieval and visualization tools. By regarding Telnet sessions as documents, which may or may not include attacks, a session that contains a certain type of attack can be used as a query, allowing us to search the data for other instances of that same type of attack. The use of information visualization techniques allows us to quickly and clearly find the attacks and also find similar, potentially new types of attacks.

1 Introduction

The proliferation of the Internet over the last few years has brought many new and improved services to the populace, but with the good, there must be the bad. There has been a new type of crime to hit the information superhighway, *Network Intrusion*. Network intrusion occurs when an unauthorized entity gains access to one or more components of a network.

The motivation behind these experiments was to develop more effective network intrusion detection tools through the combination of information retrieval and information visualization techniques. The goal of our work is to use multi-dimensional visualization to detect attempts, successful or not, at network intrusion.

Our system combines the Telltale information retrieval system and the Stereoscopic Field Analyzer (SFA) information visualization system to create an effective intrusion detection solution. Telltale is a dynamic hypertext environment that provides full-text information retrieval from a text corpus [MILL99, PEAR97]. Telltale computes the similarity between a given document and a query based on the frequencies of n-grams (n character sequences of text). SFA uses glyph-based volume rendering to visualize multi-variant, multidimensional data, enabling more complex data relationships and information attributes to be visualized than in traditional 2D and surface-based visualization systems.

In the next section, we describe our test data, and provide additional details on Telltale and SFA. In Section 3, we describe the two phases of our experiments. In Section 4 we discuss our results. Finally, in Section 5 we present our conclusions and plans for future work.

2 Background

We have explored and evaluated the effectiveness of combining information retrieval (IR) techniques with information visualization techniques as a solution to the Network Intrusion Detection problem. Below, we describe the sample network data set that we have used and the details of the IR and information visualization tools we chose for our experiments.

2.1 Data

The data that was used in our experiments came from the 1998 off-line intrusion detection evaluation (IDEVAL), which was conducted by MIT Lincoln Laboratory under DARPA sponsorship. An intrusion detection evaluation test bed was developed under this program which generated normal traffic similar to that of a U.S. government site containing hundreds of users on thousands of hosts.

The contents of network traffic such as SMTP, HTTP, and FTP file transfers were either statistically similar to live traffic, or sampled from public-domain sources. Telnet sessions were generated from statistical profiles of user types that were used to generate interactive sessions. These statistical profiles indicated the frequency of occurrence of different UNIX commands (e.g. mail, lynx, ls, cd, vi, cc, and man), typical login times and telnet session durations, typical source and destination machines, and other information [LIPP00].

More than 300 instances of 38 different automated attacks were launched against victim UNIX hosts during a simulated nine-week exercise. Attack scenarios were developed for different attackers. For example, one attacker collected information and left a back door; another was a novice hacker who broke in and then left, and a third was a disgruntled employee [CUNN99].

The following attack families were included in the evaluation: *user to root, remote to local*, *denial of service,* and *probe/surveillance*. A *user to root* attacks occurs when a local user on a machine tries to obtain privileges normally reserved for the UNIX root or super user. In *remote to local* attacks, an attacker who does not have an account on a victim machine sends packets to that machine in order to gain local access. *Denial of service* attacks are designed to disrupt a host or network service. *Probe/surveillance* attacks occur when an unauthorized user scans a network of computers to gather information or find known vulnerabilities [LIPP00], perhaps in order to then launch one of the other attacks. For a more detailed explanation and definition of these families of network attacks, see Kendall's thesis [KENN99].

2.2 SFA

The SFA visualization system is a tool for visualization of multidimensional and volumetric data [EBER96]. SFA combines glyph-based volume rendering with a minimally-immerse interaction metaphor to provide interactive visualization, manipulation, and exploration of multi-variant, volumetric data. SFA uses a glyph's location, 3D size, color, shape and opacity to encode up to nine attributes of scalar data per glyph [EBER97]. Attribute mappings can be changed in real-time, data can be filtered, and subsets can be created, allowing the user flexibility in the display of the data set.

By using glyph-based volume rendering, SFA does not suffer the initial costs of isosurface rendering or voxel-based volume rendering, while still offering the capability of viewing the entire volume. Glyph rendering also allows the simultaneous display of multiple data values per volume location. SFA allows the three-dimensional volumetric visualization, interactive manipulation, navigation, and analysis of multi-variant, time-varying volumetric data, increasing the quantity and clarity of the information conveyed from the visualization system [EBER96]. SFA has been successfully used for both scientific and information visualization tasks. We have previously applied SFA to the information visualization tasks for visualizing document similarities [EBER97] and visualizing document authorship with very successful results.

2.3 Telltale

Telltale is an IR system that provides full-text search in text corpora that may be garbled by OCR or transmission errors, or may contain text written in languages other than English. Unlike most IR systems, Telltale uses n-grams, rather than keywords or phrases. An n-gram is defined as a sequence of n consecutive characters, typically including whitespace, punctuation, and so forth. Two documents (or a document and a query) are considered similar if a sufficiently large number of the same n-grams (more than would be expected due to chance) appear in both documents. There is no notion of stemming, or stopword processing, as in word-based IR systems. As a result, n-gram based IR systems are, in general, less language-specific than other IR systems [PEAR97]. (Typical IR systems reduce the number of terms to be indexed by excluding so-called "stopwords" which appear in virtually every document and therefore have little or no discriminating power. However, the set of stopwords varies from one language to another, and is therefore a source of language dependence.)

The data being analyzed in these experiments is not ordinary natural language text. In fact, the data is drawn from tcpdump output, so there are timestamps, IP addresses, and acronyms in much greater quantity than in ordinary text. One of our main objectives was to see how well an n-gram based IR system would handle such data.

3 Experiments

The IDEVAL data set that was used in all our experiments was initially pared down to a subset that included only Telnet packets, i.e. packets that involved port 23 as either the source port or destination port. The IDEVAL data set consists of seven weeks of TCP traffic for training, and another two weeks of TCP traffic for testing. We used five weeks' worth of the seven weeks of training data, resulting in about three million Telnet packets. (Limits in our database software prevented us from using the remaining weeks of data.)

3.1 Phase 1

Initial experiments on the reduced data set involved the writing of several Perl scripts and analyzing initial processing results. These scripts created histograms on various combinations of attributes of the data. Here we defined attributes of the data to be analogous to columns in a database, e.g. timestamp, protocol, and so forth. Histograms gave us a general feel of the distribution of the data set. The most insight was gained when we used the scripts to create histograms on the combination of source IP address, source port, destination IP address and destination port. From this particular combination we were able to detect a number of *denial-of-service* attacks. This combination proved powerful in that this particular type of attack could be discovered reliably in a wide variety of situations. However, there are denial of service attacks (such as UDP floods, for example) that cannot be detected using simple histograms of tcpdump data, so from a network intrusion detection standpoint the scripts were limited. The numerous other families of network attacks still remained hidden within the corpus.

3.2 Phase 2

The experiments described above gave us a foundation for developing a robust and powerful methodology for detecting network intrusions. The chief insight was that

Telnet sessions could be regarded as documents. As a result, a corpus of tcpdump traffic possibly containing "attack" sessions can be regarded as a corpus of documents, and a session that includes an attack can be regarded as a query.

From the Telnet packets we extracted from the IDEVAL data set, we developed a procedure for reconstructing the Telnet sessions in their entirety. This conversion from packets to sessions involved creating a database to hold our network data and then developing the scripts that would extract and convert the individual sessions. The database consisted of two relations, one for the connections, and the other for the packets. A one-to-many mapping existed between the connection and packet relations. Once the database schemas had been developed, the database was created on a MySQL [MYSQL] database server loaded onto a four-node Beowulf cluster. Our Telnet packet data was then loaded onto the database server. Several Perl scripts were written to extract the sessions from the database and then convert them from their native hexadecimal format to ASCII. This extraction and conversion allowed us to analyze the data using our information retrieval tools.

Using Telltale, we calculated similarity scores based on how similar or dissimilar the sessions (documents) were to the attacks (queries). The IDEVAL data includes a list of the known network intrusion attacks (e.g. ffbconfig, dictionary, portsweep, etc.), times that the particular attacks occurred, as well as the source and destination machines on which the attacks occurred. This known set, or truth set, of network attacks allowed us to create a set of queries with known answers, i.e. we knew which attacks occurred and which sessions were involved. We created five session corpora, where each corpus contained approximately fifty sessions. In each, perhaps five or ten sessions were attacks. We assigned sessions to corpora based on size of session, and timestamp. The size of session was an important attribute because we needed to have sessions within the corpora that were comparable in size to the various attacks. If we had placed sessions distinctly different from the attack in our corpora, our results would have been skewed. Timestamps also played an important role because those sessions shortly before an attack might hold peripheral information that could be useful in detecting the impending attack(s).

We loaded each of the five session corpora into Telltale, one at a time, and used the known attacks as queries. The output from each query was a list of similarity scores, i.e. the similarity between the "attack" query and the various sessions. If, for example, there were three ffbconfig attacks, then we received three lists of scores. If the attack session itself occurred in the corpus, the normalized similarity score for that session was very high.

To visualize the relationship, if any, between attacks, we loaded the similarity score lists into SFA, using each list as its own dimension. The scores for the three ffbconfig attacks, for example, were assigned arbitrarily to the x, y, and z dimensions in SFA. Traditionally, the first three dimensions of a data set are mapped in this way, within the three-dimensional environment. With SFA, other dimensions can be mapped to such parameters as color, size, transparency, shape, and vector components. Had we had more ffbconfig attacks to use as queries, we could have assigned them to any of these six remaining dimensions. Through the SFA system interface we were able to, in real time, map our three ffbconfig attacks to different dimensions. With this system flexibility we gained a better understanding of our session corpora from multiple views of the same data. Changing the data mappings and interactively exploring the visualized data provided easier analysis of the data and enabled pre-attentive visual similarity processing and fast visual clustering.

4 Results

As expected, we found that if an attack occurred in the corpus, then we had no difficulty finding the attack session using that same attack as a query. For example, using a given ffbconfig attack as a query, we were able to find that same attack in the corpus if it was present. The most useful result was that we were also able to spot other ffbconfig attacks.

We also discovered that we were able to spot attacks that were within the same family as the query. For example, we were able to discover an eject attack when we used a ffbconfig attack as the query. These two attacks are variants of a user-to-root network attack, and in fact both are buffer overflow attacks, so the system is useful in detecting "families" of attack types. Our system should also be effective for detecting new attacks based on variants of known attacks.

Figure 1 (see Appendix) shows the results of visualizing three ffbconfig attacks as the queries against one of the session corpora. Notice the cluster of glyphs, each of which corresponds to a session in the corpus, grouped around the origin. These sessions have normalized similarity scores near zero when compared with the attack query. In the far right, front and top corners of Figure 1 we see other glyphs, which are the three ffbconfig attacks themselves. However, the attacks are not in the extreme corners of the display, as one would expect. Glyphs for each attack are attracted to the other axes, pulling each glyph slightly away from its corner.

An attack query of a given type can find itself, and other instances of that same type of attack. Furthermore, we were able to spot attacks of different types within that same family of attacks. For example, a ffbconfig attack can be used to find itself, other instances of ffbconfig attacks, *and* other buffer overflow attacks, such as eject, as shown in Figure 2 (see Appendix). Figure 2 shows the results of using three ffbconfig attacks as the queries against a session corpus. All of the sessions without an attack within this family are clustered at the origin. The remaining session glyphs contain attacks within this family. There are three ffbconfig attacks in the session corpus visible near the extreme right, lower left and upper corners. Somewhat closer to the origin, but still distinct from it, are sessions containing other buffer overflow attacks, such as eject and fdformat. The spatial location of these attack sessions also shows the similarity of each of the different ffbconfig attacks and may help determine from which known ffbconfig attack a new attack was derived. Some of these similarities can be attributed to the possibility that core pieces of programming code used by network attackers to construct these network intrusions are similar. We suspect that the similarity does not stop here; therefore, further experiments along these lines are needed.

5 Conclusion and Future Work

These results support the claim that there are in fact underlying patterns associated with different network attack families, and that these patterns can be detected and visualized. Our experience with visualizing Telnet sessions indicates that displaying attacks in a higher-dimensional space leads to insights that would be harder to come by in a two-dimensional visualization.

Our next step will be to investigate system scalability with respect to attacks that don't take place in Telnet sessions. We also plan to investigate detection of attacks in closer to real time. To do this, we will experiment with methods that add the network sessions to the corpus just after their completion. This improvement will allow system administrators to identify possible attacks by simply looking at the visual output. It may also be possible to show a system administrator results of the form: "with n% probability the following unfinished ‘session’ is an attack of type y". The use of similarity isosurfaces within the SFA display could be used as a visual cue to show probability of attack sessions. Even if we can't identify a new attack by type, it would be desirable to identify the attack by probable family.

Other experiments that will be preformed are with different policies for aging of sessions from the session corpus. Such a policy is necessary since otherwise we end up with an infinitely large session corpus. If models of certain attacks can be developed over time, it may be that detailed sessions of those types of attacks are no longer needed.

We will also explore the addition of more session details as metadata within the visualization display, and explore the effectiveness of different glyph attributes for conveying important intrusion detection session attributes. We have only explored a small portion of the potential benefit of information visualization for discovering network intrusion attacks and we will continue to refine the visualization process to more effectively highlight intrusions.

References

[CUNN99] R. K. Cunningham, R. P. Lippmann, D. J. Fried, S. L. Garfinkel, I. Graf, K. R. Kendall, S. E. Webster, D. Wyschogrod, M. A. Zissman, “Evaluating Intrusion Detection Systems without Attacking your Friends: The 1998 DARPA Intrusion Detection Evaluation,” SANS, 1999.

[EBER96] Ebert, D., Shaw, C., Zwa, A., and Starr, C. "Two-handed Interactive Stereoscopic Visualization," IEEE Visualization '96 1996.

[EBER97] Ebert, D, Kukla, J., Shaw, C., Zwa, A., Soboroff, I., and Roberts, DA., "Automatic Shape Interpolation for Glyph-based Information Visualization," IEEE Visualization 97 Late Breaking Hot Topics, October 1997, Phoenix, AZ.

[KEND99] K.Kendall, “A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems”, S. M. Thesis, MIT Department of Electrical Engineering and Computer Science, June 1999.

[LIPP00] Richard P. Lippmann, David J. Fried, Isaac Graf, Joshua W. Haines, Kristopher R. Kendall, David McClung, Dan Weber, Seth E. Webster, Dan Wyschogrod, Robert K. Cunningham, and Marc A. Zissman, “Evaluating Intrusion Detection Systems: The 1998 DARPA Off-Line Intrusion Detection Evaluation,” in Proceedings of the 2000 DARPA Information Survivability Conference and Exposition, 2000, Vol 2.

[MILL99] Ethan L. Miller, Dan Shen, Junli Liu, Charles Nicholas, and Ting Chen, “Techniques for Gigabyte-Scale N-gram Based Information Retrieval on Personal Computers,” Proceedings of the 1999 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '99), Las Vegas, NV.

[MYSQL] “MySQL Reference Manual”, http://www.mysql.com/documentation/index.html

[PEAR97] Claudia Pearce and Ethan Miller, “The TELLTALE Dynamic Hypertext Environment: Approaches to Scalability,” in Advances in Intelligent Hypertext, J. Mayfield and C. Nicholas, eds. Lecture Notes in Computer Science 1326, Springer-Verlag.

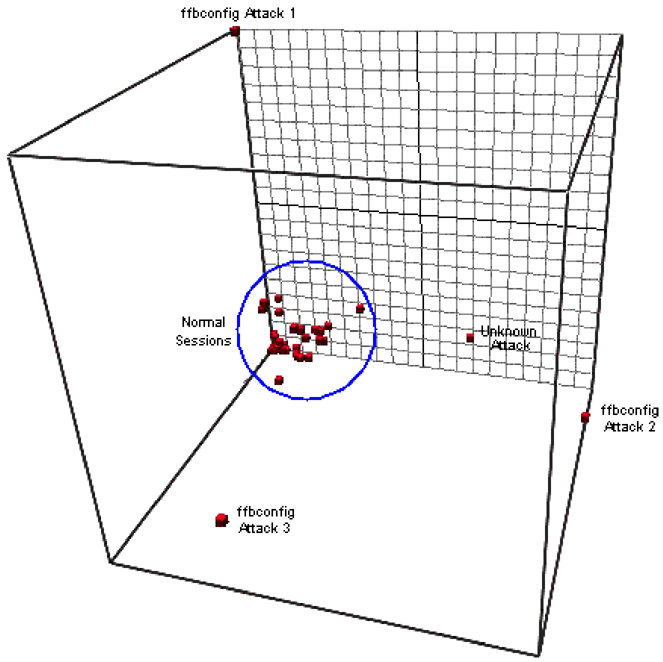

Figure 1. Discovery of the same type of attack as the query shown.

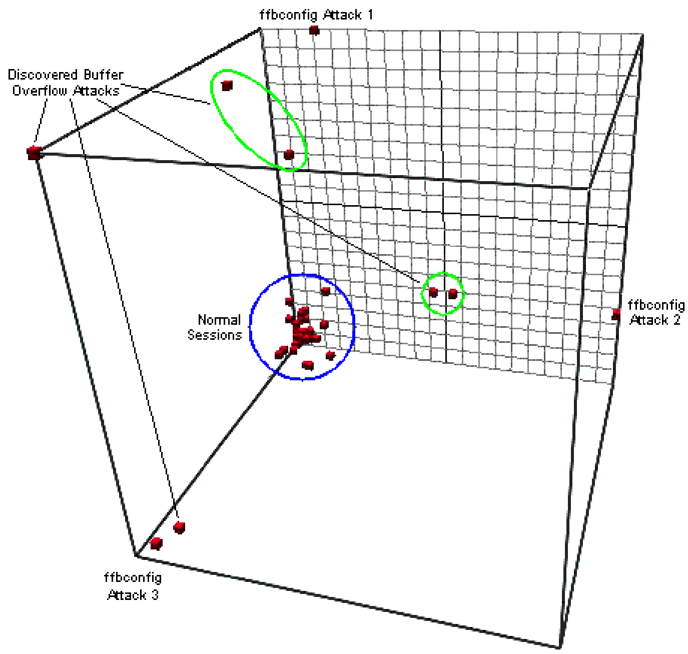

Figure 2. Discovery of the attack of the same family as the query shown.