Applications of Statistical Visualization to Computer Security

Dr. Jeffrey L. Solka, Dr. David J. Marchette, and Ms. Michelle L. Adams

This article examines the application of statistical visualization to computer security. Numerous examples are provided that illustrate the use of standard statistical visualization tools for the analysis of computer network data. It is the intent of this article to suggest ideas that might prove fruitful upon additional investigation.

Introduction

The statistics community has a rich history relating to the analysis of a disparate group of data types. Unfortunately, the community has often proceeded with this analysis well after the inception of the discipline area. Our intent has been to circumvent this trend in order to ascertain the benefit of statistical analysis—in particular statistical visualization—in the newly developing area of computer security.

Computer security and intrusion detection has existed for a number of years as a discipline. One could say, however, that the recent propagation of personal computers throughout homes, businesses, and governmental agencies has lead to a real boom for this industry. In fact, recent world events has led to an even greater focus on the safety of the informational infrastructure of the United States.

The analysis of the data sets that are associated with this domain area are particularly vexing. The data itself is often text oriented and, hence, it must be painstakingly parsed prior to any sort of statistical analysis. There are additional issues such as dimensionality and cardinality that also complicate the analysis of said data. The statistical visualization analyst is faced with the difficult task of attempting to provide "data insights" to the intrusion detection analyst that are not available from a cursory analysis of the data set.

The Netcentric Warfare (NCW) Department of Defense Report to Congress stated, "NCW is no less than the embodiment of an Information Age transformation of the DoD." In fact, George W. Bush, our Commander in Chief, stated, "We must build forces that draw upon the revolutionary advances in the technology of war…one that relies more heavily on stealth, precision weaponry, and information technologies." The report identified the "lack of secure, robust connectivity and interoperability," as one of the impediments to progress in the NCW arena.

It is likely that the attainment of this secure environment will involve human interaction with the myriad of information that is provided as part of the network defense, network information, NCW process. The visualization techniques discussed within represent a first step towards providing the human "in the loop" with a better way to visualize/understand these complex information sources.

BACKGROUND

Our initial work has focused on the application of preexisting statistical visualization methods to the analysis of network and intrusion data. The techniques discussed within include data matrix visualization, hierarchical clustering, parallel coordinates, and scatter plots. Much of this work first appeared in the conference papers, References 1 and 2. The reader is referred to Reference 3 for an in-depth treatment of the application of many of these visualization methods to computer intrusion detection and network data.

The first tool to be discussed is data matrix visualization. This tool, like much of the data visualization methodologies of recent years, first appeared in Reference 4, although more recent treatments and extensions of this idea can be found in Reference 5. The data image is based on encoding each of the individual dimensions associated with an observation in a higher dimension space as a pixel in an image. So each datum/variable combination is encoded as a pixel. In this manner, one can convert a moderately large set of observations in a fairly high dimensional space into an visual image that can be assessed for relevant patterns. We have used data matrix visualization to examine traffic to a given site at the site/machine level.[1]

The second statistical methodology that we have applied to the computer security problem is cluster analysis. Cluster analysis in general is the quest to group similar objects together. In order to perform cluster analysis, one first has to obtain some sort of encoding for the object. These encodings are usually referred to as features in the statistical clustering or pattern-recognition arena. Once the objects are encoded, then one must decide how to measure the similarity of the objects. Finally, one must decide how to ascertain the cluster structure that is inherent in the set of features that encode the object. The reader is referred to Reference 6 for a highly readable treatment of the subject area.

There are numerous ways to cluster objects. We have typically used a method know as agglomerative hierarchical clustering in our analysis. In this method, one begins with each object in a separate cluster and then, at each stage in the procedure, attempts to group together the two clusters that are most similar. This, of course, requires the ability to be able to measure similarities between clusters, and the choice of this measure can have a marked effect on the cluster structure that is ultimately discovered. We have used cluster analysis to group machines based on their internet traffic.[7, 1] We have also used cluster analysis to group a particular machine's activities on a daily basis.[2]

One is also faced with the difficulty of graphing the data associated with Internet traffic. The dimensionality of the extracted data is typically larger than three. This high dimensionality necessitates the use of nonstandard coordinate systems for displaying the data. Here again, we have turned to the statistical visualization community and employed a method that has been well known to them since 1990. This technique, deemed parallel coordinates, relies on placing the coordinate axes parallel—rather than perpendicular—to one another. In this manner, one can successfully plot high-dimensional data. Parallel coordinates like data matrix visualization was first explored in Bertin.[4] The introduction of parallel coordinates to the statistical community occurred in the 1990 paper by Wegman.[5] We have used parallel coordinates to perform a preliminary analysis of some of the network scanning tools and to draw some pictures of normal network traffic.[1]

RESULTS

Before proceeding on to the discussions of some specific results, it would probably be helpful to provide a modicum of background material on computer network traffic. See Reference 8 for a thorough treatment of the intricacies of internet

communication. Computers on the internet exchange packets of information. The exact content of the packet headers and associated packet bodies depends on the methodology, or protocol, associated with this particular type of communication. For the sake of our discussions, we will focus on the transmission control protocol (TCP)/ internet protocol (IP), or TCP/IP, method of communicating between the systems. Some of the familiar various internet services, such as email and file transfer protocol (FTP), use the TCP/IP protocol. Daemon UNIX programs run on a client system, waiting to facilitate communication between machines using the various protocols. The establishment of a TCP session between a client machine and a server machine involves the exchange of several packets with certain flags set in the packet header section of the information. Please see Figure 1, which illustrates a successful three-way handshake between two machines.

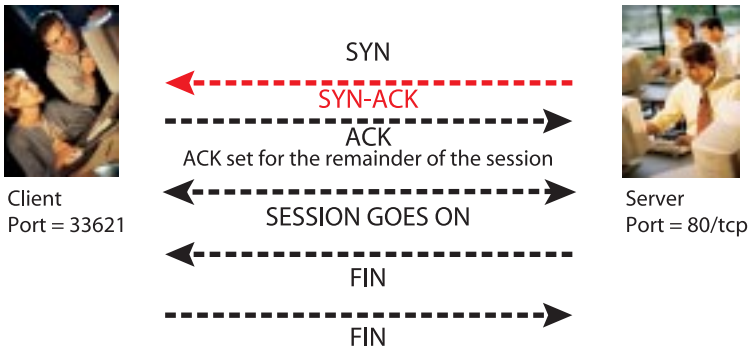

Figure 1—Cartoon of a Typical Three-Way Handshake

Each packet that travels between the two machines has a particular source port, destination port, source IP address, and destination IP address associated with it. This information is located in the header of the TCP packets. TCP uses port numbers to distinguish between (demultiplex) different logical channels on the same network interface on the same computer. The IP address of a machine is a set of four octets separated by dots, A.B.C.D. Each of the four numbers is constrained to be between 0 and 255, inclusive. They provide an address that is used by the TCP/IP protocol in its packet routing process.

With these preliminary discussions in hand, consider Figure 1. The client machine instigates a telnet session by sending a packet from his port 33621, with a destination port set to 23 and a syn (synchronous) flag set. By tradition, port 23 is the port that is usually associated with the telnet service on a machine. The packet from the client is allowed to initiate from any number of open high-numbered ports and, in this case, port number 33261 was chosen. The server responds with a syn ack (acknowledgment) from port 23. The client then responds with a packet with the ack flag set acknowledging completion of the handshake. The session then proceeds until, finally, both the client and server send a packet with the fin (finish) flag set.

Machine/Port Visualization

In our first application session, we will investigate the application of data matrix visualization to TCP traffic visualization. Figure 2 contains a data matrix plot of a month's worth of TCP traffic on a site with around 38,000 machines. Each machine corresponds to a pixel in the image. Since all the machines on this network share the first two octets of the IP address, we represent each machine as a pair (octet3, octet4). A dot is plotted if there is any syn/ack traffic associated with this third octet (x-axis)/fourth octet (y-axis) combination during the time period in question. We have taken the precaution of subjecting each of the third and fourth octet values to a transformation so that an astute examination of the figure could not reveal important details about the underlying network of machines. We have chosen to plot only those machines with 10 or more syn/ack packets during the time period. This allows us to reduce the amount of clutter associated with the plot.

A close examination of this figure indicates a "cable" that runs through the figure. This cable corresponds to one particular subnet, where nearly all of the octet4 values are occupied and had at least 10 syn/acks during the time period. An interactive system would include the capability to click on a particular dot in order to obtain more information about that dot, such as the previous traffic records associated with the machine.

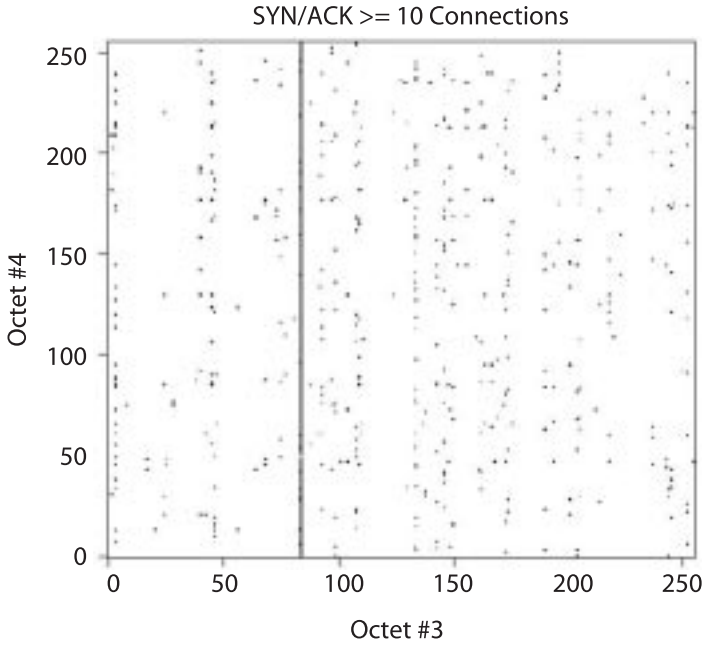

Figure 2—Pixel Image of the Syn/Ack Activity of 38,000 Machines During a 1-Month Period. Only machines with 10 or more connections during the month have been plotted.

The next sequence of figures is similar to the previous two, with the exception that they represent snapshots based on ports rather than IPs. This figure is obtained by assigning each row in the image to a sequence of ports. For example, the first row starting at the bottom of the image corresponds to ports 0–255, the second row ports 256–511. Proceeding upward in the image, one obtains the full set of 65,536 ports. Figure 3 corresponds to a data matrix image for those ports that received a syn/ack packet during the same time period as the previous two figures. One can see a concentration of dots associated with the privileged services, such as telnet and FTP, that run on ports lower than 1024. This is followed by a rather sparse region and a more highly cluttered band. Without an "on-the-fly" drill-down capability, it is hard to analyze the nature of the structure in these cluttered areas. Services such as FTP will often utilize a sequence of these higher order port numbers for data transmissions. As indicated in our previous three-way handshake cartoon, a session can be initiated from any of a number of higher order ports.

Before leaving this section, it would be helpful to provide a little more guidance as to how the graphical tools illustrated in Figures 2 and 3 could be generally useful. First we note that the method illustrated in Figure 2 could provide a human operator with a general indication of the amount of traffic at his site. Similarly, the technique illustrated in Figure 3 could be used as an indicator of the types of applications that are active on the network. These types of tools could serve as "dashboard"-type components, with an associated drill-down capability offered.

Activity Vector Analysis

In this section, we will describe an application of statistical cluster analysis to site vulnerability analysis. At many sites such as NSWCDD, each machine is accredited to run certain services on the machine. In reality, however, individuals often intentionally or unintentionally change the services that are running on machines at a particular site.

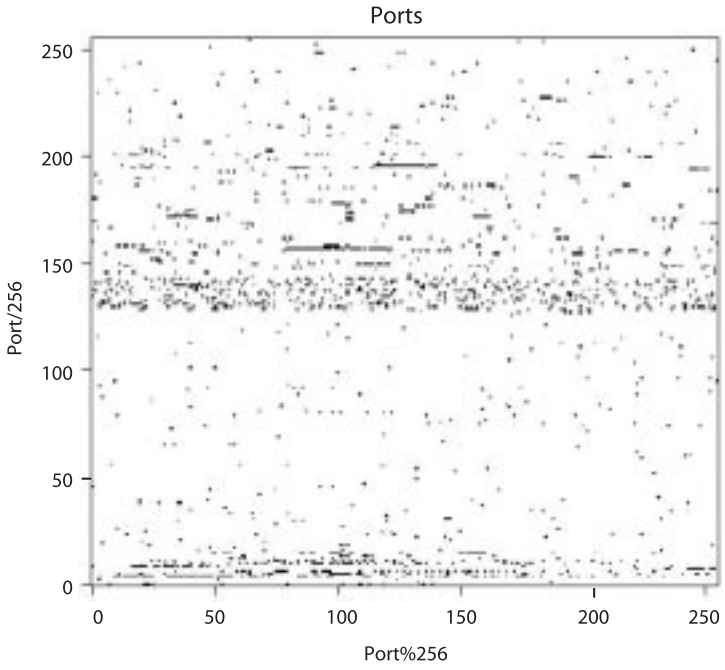

Figure 3—Pixel Image of the Port Activities of the Data Set of Figure 2

To first order, the services that are running on a given machine are characterized by the ports responding with syn/ack packets on the machine. So one could use the number of syn ack packets to each port on a machine as a "feature" that characterizes the activity of the machine. The number of packets at each port within a period of time, or equivalently the probability of accessing each port, could be used to perform a hierarchical cluster analysis, as described above, on the machines at a particular site. One would, of course, need some method to measure the distance between feature vectors, and we have chosen to use standard Euclidean distance for this purpose.

The experiment to be discussed consists of an analysis of TCP data assembled on 993 machines. This is old data consisting of roughly 1.7 million packets to 668 different ports. Figure 4 displays the dendrogram obtained via the application of a standard agglomerative clustering procedure, with complete linkage clustering to a subset of the data. The complete linkage clustering method measures the distance between two clusters as the farthest distance between them. We have labeled the terminal leaves in the dendrogram with the service that features prominently in this particular machine/cluster. For example, one can see that the first cluster of machines is primarily engaged in FTP traffic, and that the last cluster of machines is engaged in email traffic. The reason for multiple clusters of email is the difference in the amount of traffic and the other services running on the machines.

Once one obtains candidate clusters, one is faced with the task of trying to decide why the particular machines ended up in the cluster together. Hence, one needs some way to examine the activity vectors for the machines in a given cluster. In Figure 5, we present a dotplot of the network activity vectors associated with cluster 33. The x-axis presents the log base 2 of the count vectors, and the y access represents the given port. So the activity vector for the machine encoded with the triangle symbol (with its point up) has very high traffic on port 21, with virtually no traffic to any of the other ports that were active in this cluster. The dotplot can be used to study the traffic resident in a particular cluster, assuming that there are a small number of machines in the cluster.

Now that we have provided a general description of clustering activity vectors, a few words about the application of this approach are in order. As discussed earlier, each of the machines at our site is subjected to an accreditation process. One must indicate during this process the types of services, ports that are open, on the machine. For example, if one were planning to configure the machine as a web server, then this type of information would need to be indicated as part of the accreditation process. Our clustering approach should be able to group together those machines that are associated with particular types of services. Another application of this method would be to identify machines that have encountered a large change in activity level or else a

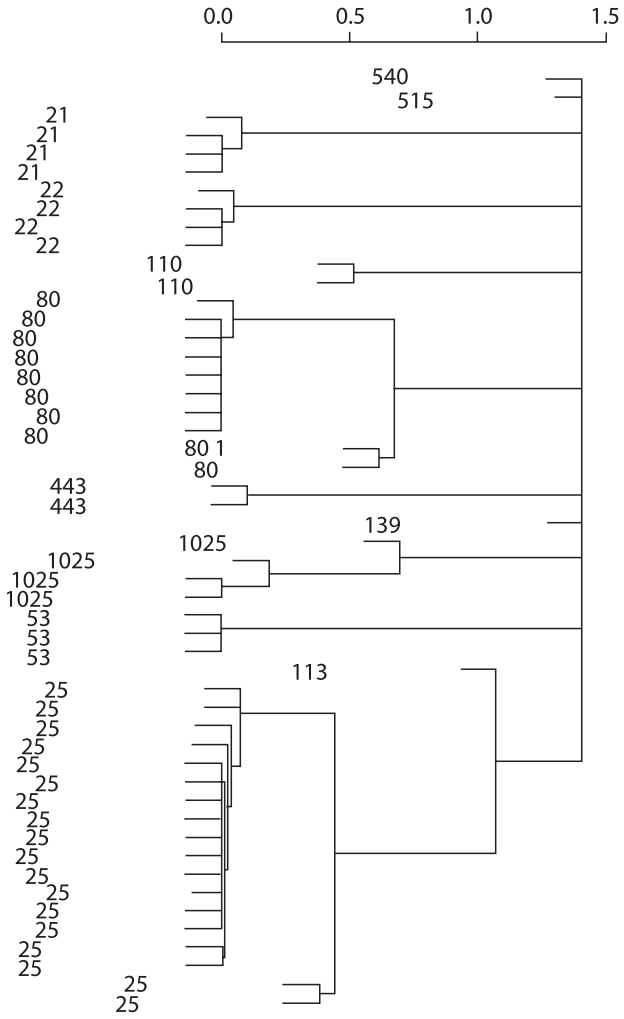

Figure 4—Dendrogram of Port Activity Vectors for a Subset of the 993 Machine, 668 Port Syn Ack Traffic

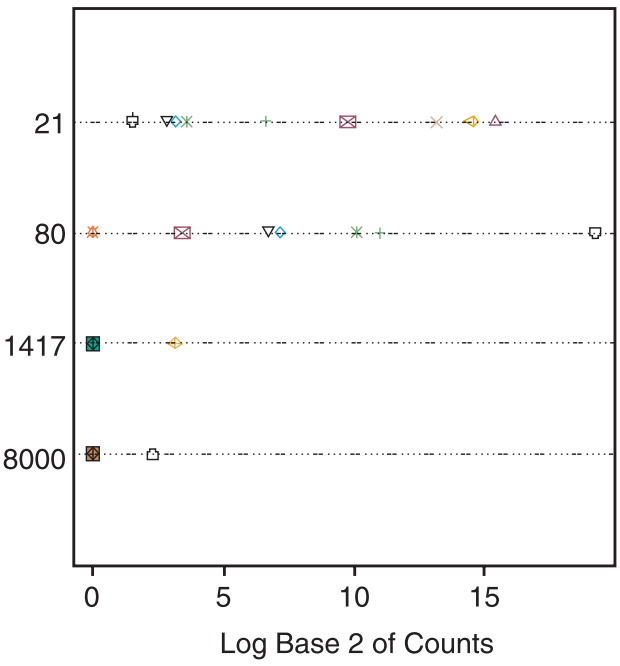

Figure 5—This is a dotplot of the activity vectors for one of the clusters obtained from a hierarchical cluster analysis of the port activity data.

change in the services associated with the machine. This type of change could be revealed by periodically reassessing the cluster structure associated with the network. Another application would be the identification of abnormal or outlier machines. A final application would be the identification of groups (clusters) of machines that may have been compromised by similar exploits. This would be revealed by the same unique port or group of ports being active on the machines in the group. This type of detection process would be defeated by those Trojans that attempt to communicate on well-known ports, although their presence might be detectable based on changes in activity level on the well-known port, as discussed above.

General Traffic Visualization

TCP and Telnet

In this section we will discuss the application of parallel coordinates to the display of network traffic. We recall that a TCP packet between two machines is characterized by time, source IP address, source port number, destination IP address, and destination port number. The use of parallel coordinates is relevant, in that each packet can be conceptualized as a point in a five-dimensional space. Visualization of time blocks of "normal" traffic is one of the simplest things that one could do with parallel coordinates. In Figure 6, we present such a plot for 1-hour's worth of TCP data. One packet is plotted as a sequence of lines that travel from the time axes to the destination port axis. The lines have been colored by source IP address. One is immediately struck by the overplotting problem that exists with such a plot. Parallel coordinates suffer from this problem, which is not necessarily surprising, in that a point in Cartesian coordinates gets mapped to a line in parallel coordinates. This mapping certainly increases the amount of ink that is associated with the plot.

One can cull the information contained in this plot by limiting the packets to a particular type of

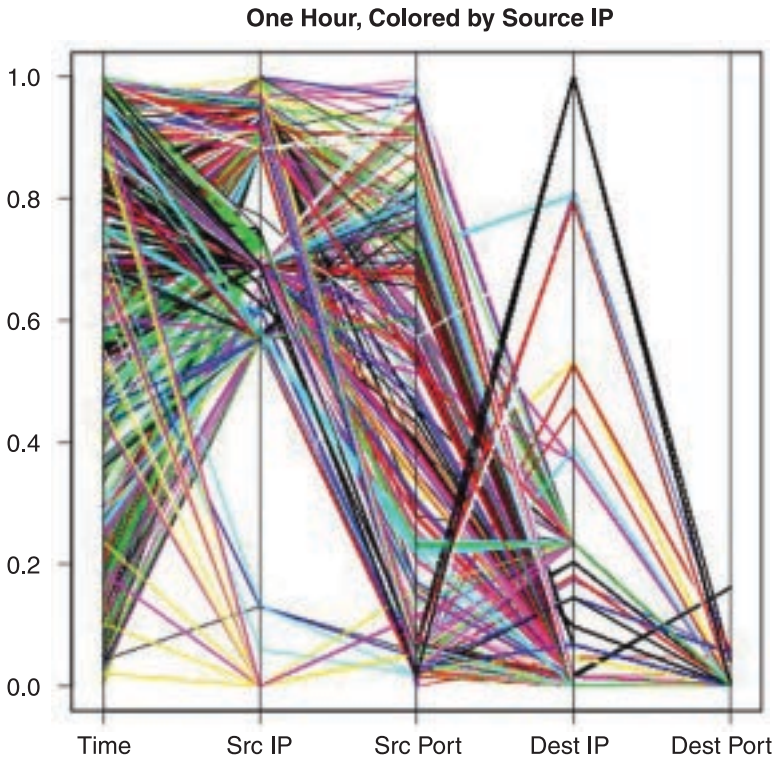

Figure 6—A Parallel Coordinates Plot Associated With1-Hour's Worth of TCP Traffic (It has been colored by source IP address.)

service. In Figure 7, we have limited the packets plotted to those associated with telnet accesses to a group of machines that occurred during a 1-hour time period. All of the lines have the same destination port coordinate of 23, since all of these packets are associated with telnet traffic. In this case, it is possible to ascertain which machines (destination IPs) seem to have the most traffic associated with them and which machines (source IPs) seem to be instigating the largest number of sessions. It is, however, difficult (if not impossible) to tell whether a particular set of lines is associated with one machine's session that has temporal persistence or a number of shorter sessions by the same machine. These plots illustrate some of the inherent drawbacks of parallel coordinates.

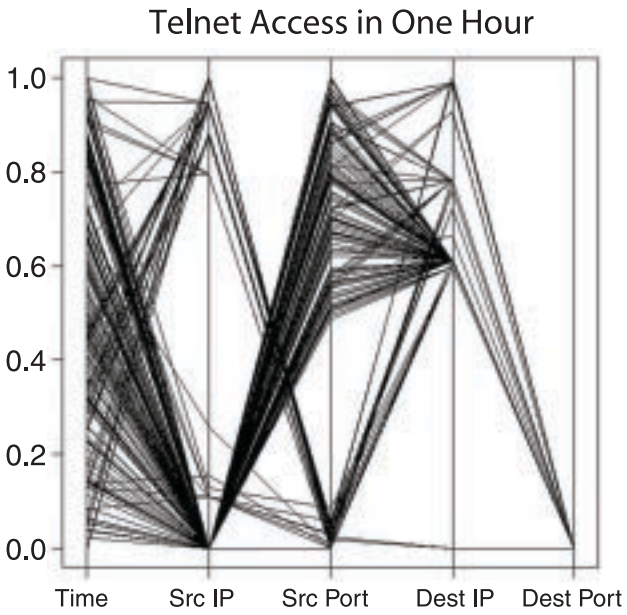

Figure 7—A Parallel Coordinates Plot of 1-Hour's Worth of Telnet Traffic

Machine and Port Scans

In this parallel coordinates example, we illustrate the application of parallel coordinates to the visualization of port-scan traffic. A port scanner is a method employed to gain information about the services and types of machines that are running on a network. Sophisticated scanners possess the capability to spoof the IP address that is associated with a particular packet and to use the information obtained during the scanning process to ascertain the operating system of the scanned machine. We present in Figure 8 a parallel-coordinates plot of a port scan of two machines. The scan was conducted from one particular machine against two other machines. We have colored the packets based on the destination IP so that each of the scans can be easily discerned. It is interesting to note that this particular scanner does not just scan each and every port on each of the two machines. In fact, the scanner relegates its scan to look only for particular ports that offer certain services. This pattern can be used to identify the specific software performing the scan.

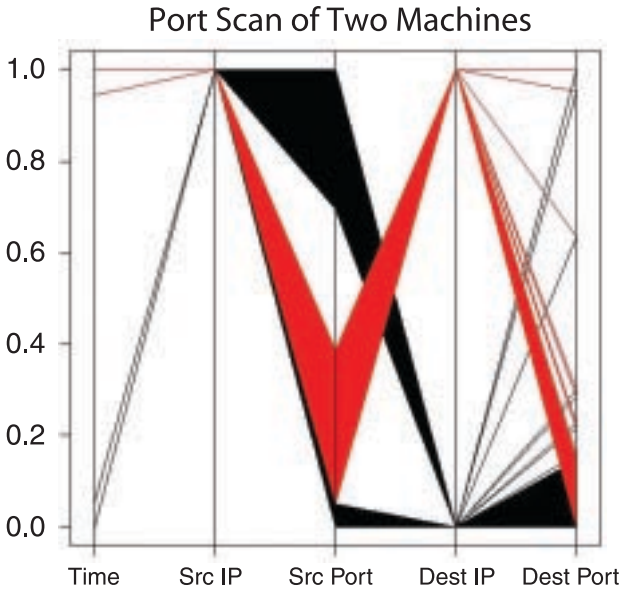

Figure 8—A Parallel Coordinates Plot of a Port Scan of Two Machines

Sessions

In this section, we focus our attention on the visualization of session information. Our first data set consists of the number of sessions for each of the machines at our site during a 1-month period. As discussed above, we may identify each of the machines by its (octet3,octet4) pair. In Figure 9, we present a plot of the number of sessions for each of the machines at our site. We have plotted a line whose length is given by the $\log_{10}$ (number of sessions) at each (octet3,octet4). The line originates at the (octet3, octet4,0) and terminates at (octet3, octet4, $\log_{10}$ (number of sessions)). The lines for the most active six sites have been colored according to

the scheme red, blue, green, cyan, magenta, and yellow. These lines have also been topped with an asterisk (]) rather than a period.

A site administrator is often interested in the most active machines at his/her site. The plot illustrated in Figure 9 provides a convenient presentation of

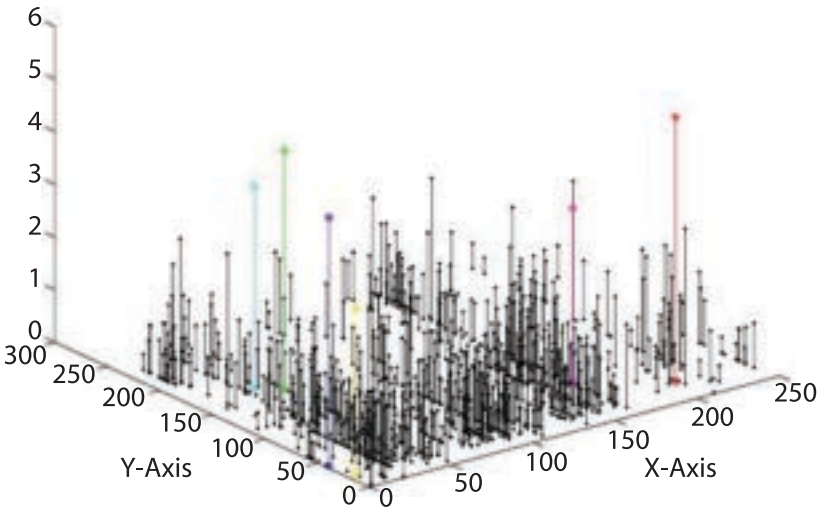

Figure 9—Log_{10} (Number of Sessions) as a Function of Octet 3 and 4 for the Machines at Our Site During a 1-Month Period of 1999

this information in terms of sessions. Alternatively, one could make the plot contingent on the number of incoming or outgoing packets. In any of these cases, an administrator needs to ascertain why the machines at his site show up in the top six in terms of activity level. Some of these machines may be the usual web or mail serve at the site, but the presence of other machines may indicate nefarious activity.

In the next plot (Figure 10), we examine the relationship between services offered and IP address for the same data presented in Figure 9. Each horizontal line in the plot represents the services that are being offered at that particular IP address. We notice that most of the machines are only running one service. We also notice that the plot does not indicate any machines that were involved in traffic on unusual ports during this time period. This does not necessarily indicate an absence of Trojan activity, in that Trojans may utilize known ports as transmission mechanisms. However, it is an indication of healthy activity on the network. Deviations from this pattern may be evidence of suspicious activity, and should be investigated.

Next, we consider a single day in August 1999 and consider the completed incoming sessions. Intuitively, it seems obvious that there is a direct relationship between the number of packets in a session and the amount of data transferred, and we explore this with a simple plot in Figure 11. Note that, contrary to what one might have guessed, there is not a single linear relationship between the number of packets and the data but, rather, there are at least two such relationships! This relationship does not appear to depend on the application (indicated by the destination port and depicted by plotting symbol), nor is it related to the time of day of the connection.

In order to further investigate this relationship, we zoom in to the figure, which is depicted in Figure 12. As can be seen, there is, in fact, somewhat of a relationship between application and the slope of the line; furthermore,

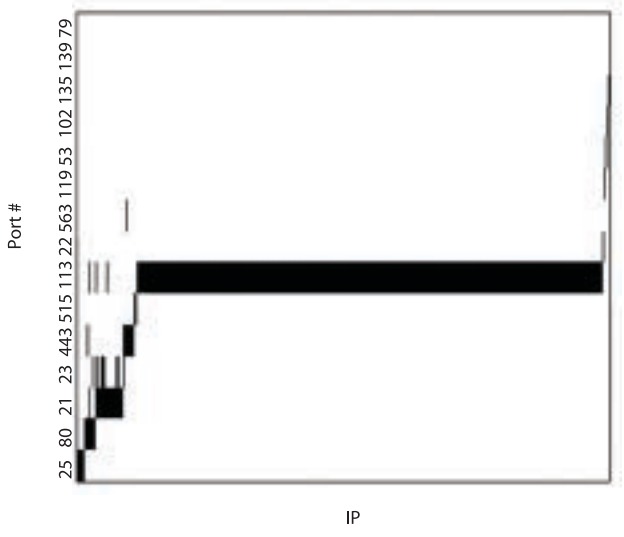

Figure 10—Services as a Function of IP Address for Machines at Our Site That Had at Least One Session During a 1-Month Period of 1999

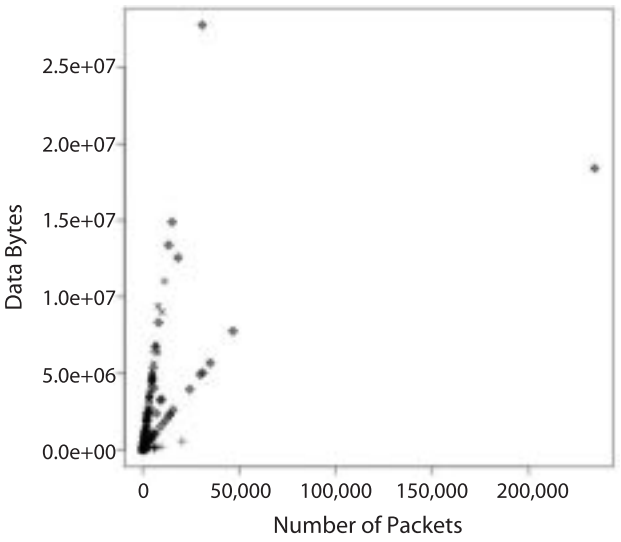

Figure 11—Number of Packets Plotted Against Amount of Data for Incoming Completed Connections. The destination port is indicated by the plotting symbol. This plot represents 8516 connections.

there are more than two lines. This kind of analysis can help to characterize normal sessions for the different applications. Note that one application can have different lines in this plot, indicating that there are different "modes" of normal activity. This leads to better models of "normal" activity.

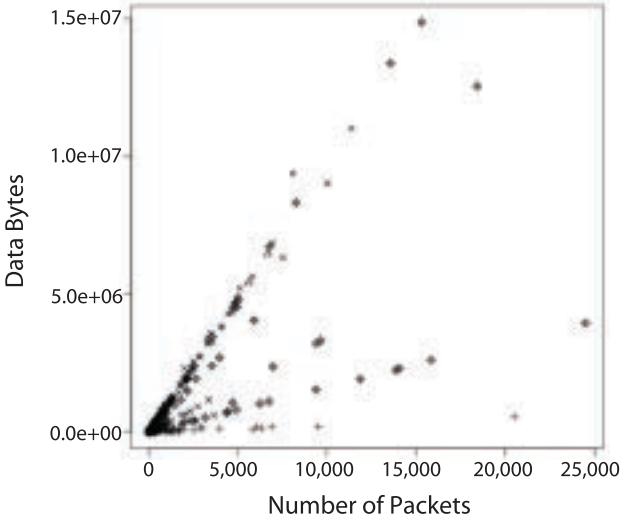

Figure 12—(Figure 11 zoomed in) This plot represents 8510 connections. The plotting symbol indicates the destination port (application) of the session.

Another view of network data is presented in Figure 13. Here we have plotted source port against time for one week of data. The interesting thing about this plot is that while we have not indicated the source IP address, it is easy to detect individual machines by the curves in the plot. For each connection, a machine selects a new (sequential) source port, and so a curve in the source ports indicates (with a high probability) that all the sessions are from the same machine. This indicates machines that have a relatively long-term relationship with our machines, such as extensive FTP or web surfing, rather than a few single sessions that may be indicative of low-level probes. Since these are completed sessions, these lines don not indicate scans, which typically consist of a large number of uncompleted sessions.

This alone might not be that interesting, but consider Figure 14. Here, we are considering a subset of sessions consisting of sessions of very few packets. The color/symbol combination is unique for each source IP. We see that the linear patterns do, in fact, correspond to single IPs. However, there is another interesting feature discernible in this graph. Note that the curves stay within very well-defined bands, and there are a number of distinct bands. These correspond to choices that the operating system makes, and therefore can be used to provide a level of passive operating-system fingerprinting.

Graphics can also be used to understand the behavior of malicious code. A Trojan program, SubSeven, has the interesting property that it scans for copies of itself, for the purposes of applying an upgrade. Machines scanning for the SubSeven Trojan have been detected at a site, and it was of interest to determine whether the same machines were scanning over and over, or whether the Trojan is smart enough to give up once it has failed to detect copies of itself. A view of the scans is depicted in Figure 15. The axes correspond to time and source IP address, with the IP addresses sorted so that IPs with similar behavior are together. As can be seen in this plot, the typical behavior is to scan for a short time period, then stop, although there are several IPs that scan throughout nearly the entire time range. This raises the question of whether the IPs that stopped did so because the Trojan was detected. This is probably

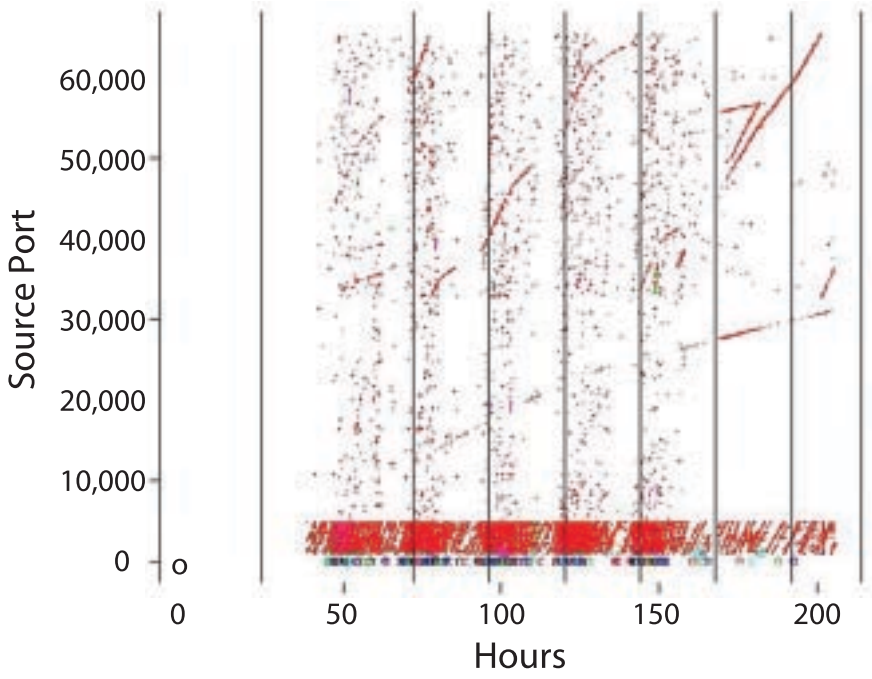

Figure 13—Source Port Plotted Against Time for 1 Week. Vertical lines correspond to days.

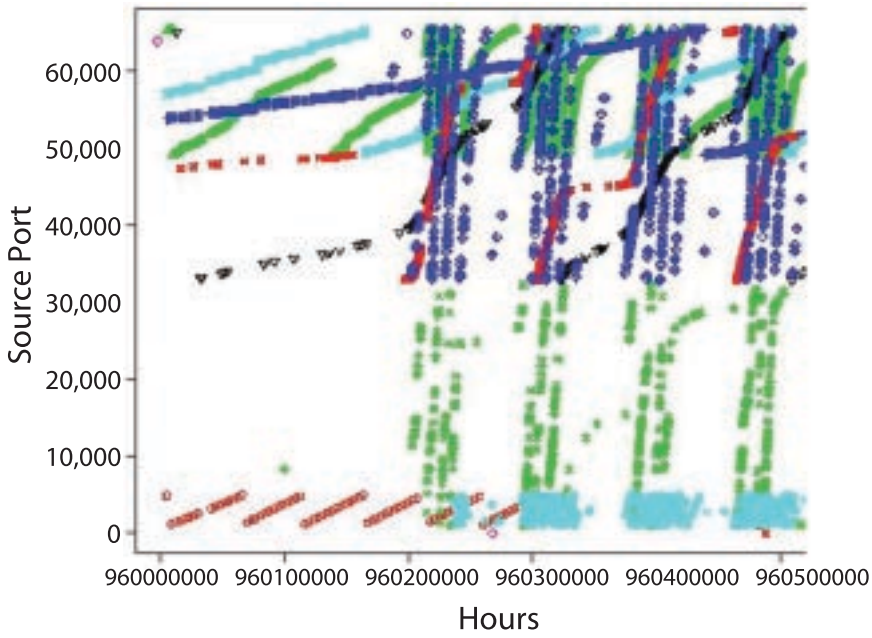

Figure 14—Source port plotted against time for sessions consisting of very few packets. The x-axis corresponds to time (seconds since Jan 1970), while the y-axis corresponds to source port. Distinct color/symbol combinations indicate distinct source IP addresses.

Figure 15—SubSeven scans, indicating that very few IP addresses ever repeat their scans. These data represent over 6 months and 2 million scans. The x-axis corresponds to time and the y-axis to IP address.

not the case due to the uniformity of incoming times for these IPs.

Another application of parallel coordinates is shown in Figures 16 and 17. One day's worth of incoming completed sessions (excepting email, port 25) is depicted in Figure 16. Each axis has been scaled in this plot to values between 0 and 1. The axes plotted are time, source IP, destination IP, source port, destination port, number of packets in the session, number of data packets in the session, and total number of data bytes transferred. One can see a measure of correlation between the number of packets, the number of data packets, and the number of data bytes. The heavy-tailed behavior of these attributes is also apparent.

Figure 17 shows the subset of the data corresponding to those sessions with destination port > 1024, which are not part of an FTP data transfer. As can be seen in this plot, each source IP goes to a distinct destination IP and a distinct port (application) at that destination. So these sessions correspond to very specific activities between pairs of computers. These services correspond to Lotus Notes, Prospero Data, and an unassigned port. This latter probably indicates an application that has been configured to run on an unused port.

Note the activity of the second source IP from the top. This computer initiated sessions throughout the day but only used a very tight band of source ports. This is an indication that either this machine is not interacting with many other machines during the day, or that its operating system restricts its source-port range to this region. A similar observation can be made about the low-value source ports, connecting to the lowest value destination port. This can provide information about the operating system of the source computer which, in turn, may be used to detect spoofing or hijacking.

SUMMARY AND CONCLUSIONS

We have explored the application of statistical visualization methodologies to the problem of the visualization of computer network traffic. Some of our

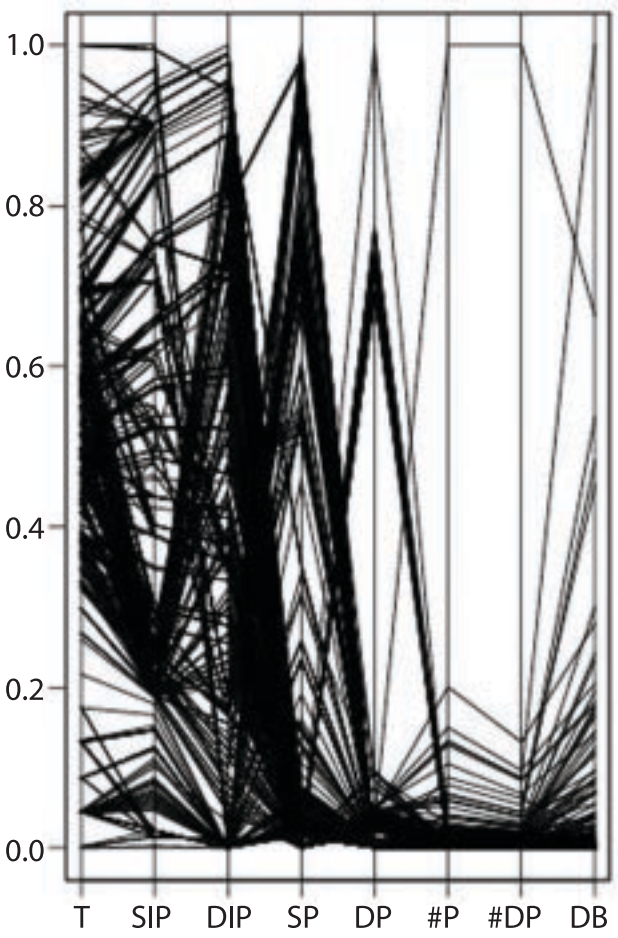

Figure 16—All the completed incoming sessions (excluding port 25 (email)) for a single day. The axes are time, source IP, destination IP, source port, destination port, number of packets in the session, number of data packets in the session, and total number of data bytes transferred. The axes have been independently scaled.

examples have focused on the visualization of data that is directly relevant to the computer security mission. We have discussed the application of data matrix visualization, hierarchical cluster analysis, dotplots, and parallel coordinate plots.

Visualization methods can help a human analyst to understand normal behavior. It can be used to detect anomalies in seemingly normal traffic. It can be used to reveal subtle patterns that may be resident in the data. These patterns can, in fact, be used to help improve models that we might build for the data.

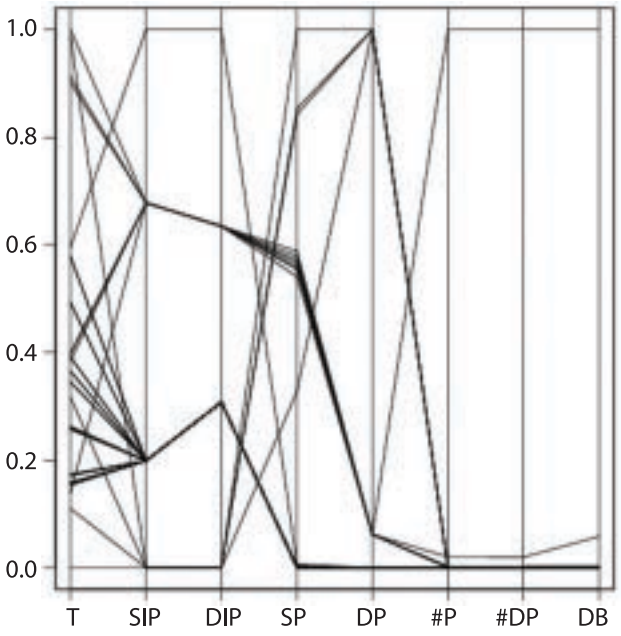

Figure 17—Completed incoming sessions to ports > 1024 (excluding source port 20 (FTP data transfer)) for a single day. The axes are time, source IP, destination IP, source port, destination port, number of packets in the session, number of data packets in the session, and total number of data bytes transferred. The axes have been independently scaled.

Much work remains to be done in this area. Ultimately we are seeking to develop techniques that will aid the human operator in his analysis of this data. We would like to be able to provide the human operator with the capability to discern patterns within the data that are not readily available from a cursory screening of the data. This will be the focus of many of our future investigations.

ACKNOWLEDGMENTS

The authors wish to thank the Office of Naval Research and the Ballistic Missile Defense Organization for their continued support of these efforts.

REFERENCES

1. Solka, J.L.; Marchette, D.J.; and Wallet, B.W., "Statistical Visualization Methods in Intrusion Detection,"

presented at and appearing in the *Proceedings of Interface 2000*, 2000.

2. Solka, J.L. and Marchette, D.J., "Functional Analysis of Computer Network Data," *Proceedings of the 33rd Symposium of the Interface of Computer Science and Statistics*, 2001.
3. Marchette, D.J., *Computer Intrusion Detection and Network Monitoring: A Statistical Viewpoint*, Springer, New York, 2001.
4. Bertin, J., *Semiology of Graphics*, University of Wisconsin Press, Madison, Wisconsin, first edition, 1983.
5. Wegman, E.J., "Hyperdimensional Data Analysis Using Parallel Coordinates," *Journal of the American Statistical Association*, 85:664–675, 1990.
6. Everitt, B.S., *Cluster Analysis*, John Wiley and Sons, New York, third edition, 1993.
7. Marchette, D.J., "A Statistical Method for Profiling Network Traffic, *USENIX Workshop on Intrusion Detection and Network Monitoring (ID '99)*, pp. 119–128, 1999.
8. Stevens, R., *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley, Reading Massachusetts, 1994.

THE AUTHORS

DR. JEFFREY L. SOLKA

Dr. Jeffrey L. Solka earned a B.S. degree in mathematics and chemistry and an M.S. degree in mathematics from James Madison University in 1978 and 1981, respectively. He earned an M.S. in physics from Virginia Polytechnic Institute and State University in 1989 and his Ph.D. in computational sciences and informatics (computational statistics) at George Mason University, working under the direction of Professor Edward J. Wegman, in May 1995. Since 1984, Dr. Solka has been working in nonparametric estimation and statistical pattern recognition for the Naval Surface Warfare Center, Dahlgren Division, Dahlgren, Virginia. His current research also includes latent class discovery, effect of metric space choice on discriminant analysis and clustering, region of interest identification in imagery and video, and statistical methods in intrusion detection. He has published over 100 journal, conference, and technical papers, and holds four patents. He has also won numerous awards, including the 1995 Independent Research Excellence Award, the 1995 Outstanding Dissertation in Statistics Award, the 1999 Office of Naval Research Special Act Award, the 1999 Award of Merit fro Group Achievement as part of the Quick Strike Planner Development Group, and the 1999 Award of Merit for Group Achievement as part of the Secondary Heuristic Analysis for Defensive On-Line Warfare (SHADOW) team. Since 1997, Dr. Solka has been a part-time assistant professor at George Mason University's School of Computational Science, as well as the graduate program coordinator for their information technology and computational sciences and informatics programs.

DR. DAVID J. MARCHETTE

Dr. David J. Marchette received a B.A. in 1980 and an M.A. in mathematics in 1982 from the University of California at San Diego. He received a Ph.D. in computational sciences and informatics in 1996 from George Mason University under the direction of Professor Edward J. Wegman. From 1985–1994 he worked at the Naval Ocean Systems Center in San Diego doing research on pattern recognition and computational statistics. In 1994 he moved to the Naval Surface Warfare Center in Dahlgren, Virginia, where he does research in computational statistics and pattern recognition, primarily applied to image processing, automatic target recognition, and computer security. He is the author of a book on computer intrusion detection and numerous journal articles.

MS. MICHELLE L. ADAMS

Ms. Michelle L. Adams is originally from Lynchburg, Virginia, where she attended Heritage High School and the Central Virginia Governor's School for Science and Technology, graduating in June 1995. She then graduated in May 1999 with a B.S. in biology from William and Mary with a minor in mathematics. In Spring 2001, she graduated with an M.S. in computer science with a specialization in computational operations research, also from William and Mary. Ms. Adams began work at NSWCDD in July 2001.