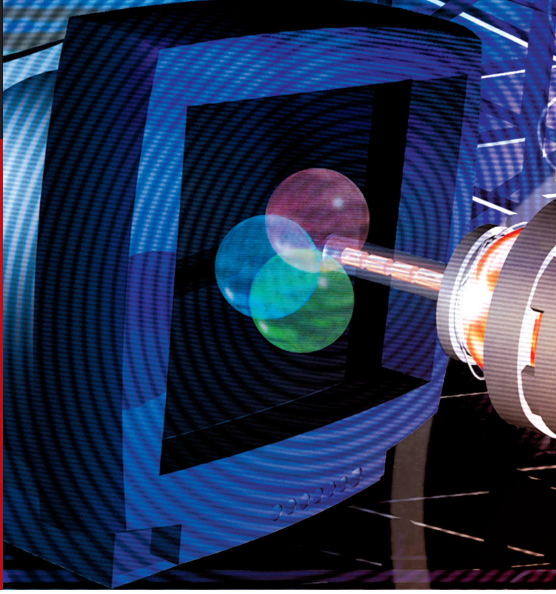
Processing Data to Construct Practical Visualizations for Network Security



by Kulsoom Abdullah, Chris Lee, Gregory Conti, and John Copeland

Network vulnerabilities are increasingly rampant despite advances in Intrusion Detection Systems (IDSs) and Intrusion Prevention Systems (IPSs). Even as funding and work by government, industry, and academia to counter these vulnerabilities increases, over 1,000 variants of worms and viruses have been discovered during the past six months [1], and the level of network traffic increases as capacity increases. [2] Network monitoring systems are already choked performing packet analyses for large networks, and traffic increases worsens the problem. [3]

Information visualization methods deal with large datasets and provide far more insight and understanding to a human analyst than viewing text alone. [4] When techniques of information visualization have been applied to the network security domain, studies have shown a significant decrease in the time required to determine many types of network threats. The use of visualization with network data to aid in security is growing, but more work is still required. This article describes methods developed to scale a large amount of network data into meaningful visualizations for intrusion detection. These techniques were incorporated into the design and implementation of a tool to facilitate log analysis for IDSs. Capturing network traffic, the tool's design, the data-scaling method used before plotting, and definitions and illustrations of several threat models will be discussed.

Capturing and Parsing Network Data

Tcpdump, a standard packet-capturing tool, collects network data, and the parameters used for visualization are then parsed from the network packet headers. The advantage of parsing network packets, compared to traffic-flow information, is that real-time processing on network packets can be performed instantaneously without having to wait for a flow to end compared to analyzing flow statistics. In our system, packet headers are parsed for information, but not the payload of the packet. This design choice was made because processing each packet payload would greatly increase the processing burden on the monitoring system.

During the design of our system, we considered requirements for both forensic analysis and real-time traffic monitoring. Forensic analysis is used on static network captures after an incident has occurred. This is often performed by browsing through the capture logs with tools such as Ethereal [5] and is considered a tedious process. Currently, we have used forensic Honeynet traffic captures from the Georgia Institute of Technology network [6] and the Honeynet Scan of the Month [7], because they provide a good benchmark to test the effectiveness of the tool.

Tool Description

A good visualization provides an overview of data by which to understand context and then provides more detail on demand. The data should be scaled and presented so that when an overall view is given, there is as little occlusion as possible in that view. Plotting data over time will show patterns and trends. Cumulative port statistics will show port activity.

Histograms are used because they are easy to interpret and good for visualizing large datasets. [8] Values can be compared, which is useful in visualizing time patterns. For three-variable plotting, we use 2D stacked, rather than 3D, for lower program complexity and processing and for more accurate value interpretation. In 3D, it is difficult to accurately determine values, as 3D is represented on a 2D surface, and this can permit an inaccurate perception. [4] The variables plotted on the graph are time, port count, and port number (or range) as illustrated in Figure 1.

Preprocessing the Data

There are many network data parameters, and some of these variables have a large range of values. Because of this, the data must be scaled before it is plotted. In the overall graph, overlap and occlusion should be avoided to reduce confusion. Network traffic statistics are highly variable by nature. High values can skew the scale and hide values that are much lower. (For a comparison, see Figure 5 and Figure 6) To deal with this, cube root instead of a logarithmic scale is used to scale data quantities, because cube root can be applied to values from 0–1 and scaled to

positive values. We can also complement the scaling with information visualization best practices, such as filtering, zooming, and mouseovers, to deal with occlusion.

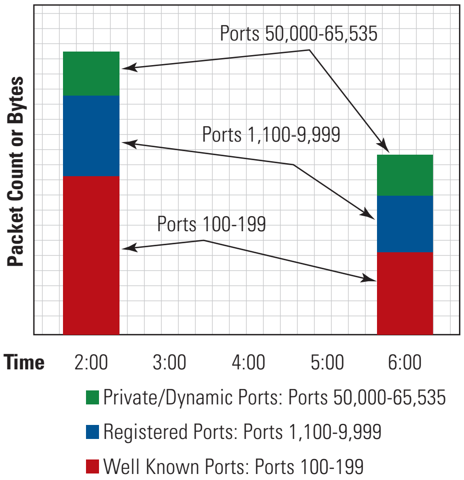


Figure 1. Layout of the Visualization. The *x* axis represents time, while the *y* axis is an interval quantity of port count or total port bytes. Port-number ranges are grouped and mapped by color.

Port Scaling

There are 65536 possible port numbers, which makes it impossible to allocate each discrete value to one pixel on an axis. Port numbers have been grouped into ranges so that we can fit the range on the graph without losing context.

The well-known and commonly assigned ports (0–1,023) are grouped into bins of 100 per group. Most attacks start with these ports, which require more granularity in this range, and, because of this, we chose smaller groupings. The registered ports 1,024–49,151 can be used by an application or be connected to a server. This range is not as active as the well-known ports, so larger groups of 10,000 are used. Typically, no service should be assigned in the private or dynamic ports range (49,152–65,535), but these can still be used by malicious applications. (See Figure 1) These ports are divided into larger groups of 40,000–49,999 and 50,000–65,535. (The plot shown in Figure 1 is illustrative but not a plot of real activity for that port range.)

Singling out individual ports is a way to filter the graph. In Figure 5, the targeted ports of that time are separated from the other port ranges. These were chosen because a Honeynet traffic capture was used. In a regular network, ports that are used most of the time would be separated. This helps filter out high port counts from the other port ranges, which could drown out other possible anomalous activity that could be occurring in its respective port range.

Time Scaling

Sampling rate and graph-update rate influence what kind of information is revealed in the data. A small time sample is good for quickly occurring activities such as fast network scans, Denial of Service (DoS) attacks, and fast- propagating worms. (See Figure 2 and Figure 3) A large time sample is better for viewing slow network scans and overall network trends over a long period of time.

Time is more crucial with real-time monitoring when activities happen quickly. However, with a time interval that is too small, too much detail may result, making it difficult to notice a pattern.

Internet Protocol (IP) Address Scaling

Like port scaling, it is not possible to plot the four billion potential Internet Protocol (IP)addresses without filtering or scaling. A matrix method has been used in SnortView [9] and NVisionIP [10] to layout IP addresses across two perpendicular axes. VizFlowConnect [11] filters on a host and maps IP addresses on a parallel plot axis. Currently, we do not have IP address information in our tool, but we are considering filtering on IP addresses that actively connect to the local network.

Results

We use typical attack captures from the Honeynet to show the effectiveness of our methods.

Network Scanning and Mapping

A scan is more difficult to detect when performed on a network's commonly used ports. When the scan probes unused IP addresses and ports, this is clearer on the graph, because those ranges were never used before and would "be readily apparent."

Figure 3 shows 30 minutes of network scans. The popular network mapping tool (nmap) was used to perform Synchronize Flag (SYN), NULL, XMAS, and Transmission Control Protocol (TCP) Connect scans of all ports. The pattern of the way in which port ranges are targeted can be seen.

Viruses, Worms, and Trojans

Distributed Denial of Service (DDoS) attacks are on the rise. Some occur when illicit Simple Mail Transfer Protocol (SMTP) servers are installed on a compromised host for Spam and Unsolicited Commercial E-mail (UCE), and some shut down other systems and the services they provide. Typically a DDoS attack is setup to compromise machines and gain control over them. Once control is established, those machines can be used to carry out attacks. In the graph, we would see higher traffic on the port/service being hacked, and then afterwards we would see activity on backdoor ports, which would be used to scan the other hosts and transmit and receive traffic. (See Figure 4)

Backdoors and Rootkits

A botnet attack capture is used to illustrate the result of a successful takeover. We see an increase of traffic on those ports opened for use. For the most common botnets, the ports typically used are 6667 or any from 6660–6670.

In Figure 5, activity can be seen on ports 80 Hypertext Transfer Protocol (HTTP), 139 Network Basic Input Output System, (NetBIOS), 445 Server Message Block (SMB), 1434 Slammer/Microsoft Structured Query Language (MS SQL) Monitor, 4899 Remote Administration Tool (Radmin), and 28431 (Unallocated). The attackers exploited port 445 and installed a program that created an encrypted backdoor port on port 4899. They subsequently compromised Honeynet machines and then added their Internet Relay Chat (IRC) botnet. In the last part of the graph on the right, we can see the successful botnet traffic on the IRC ports (light blue), which shows consistent network activity.

Figure 6 is a port-count graph of the Honeynet scan of the month. The botnet installation and subsequent botnet activity had a large number of packets transferred in and out of the network, thereby increasing the scale of packet counts and hiding the other port count values.

Conclusion and Future Work

This tool is has proven useful for detecting malicious activities that affect ports and for providing an effective overview of all port usage on a network. The tool can be used to determine anomalous behavior with an IDS and in situations in which human visual analysis can be used with anomaly-based algorithms and known signatures.

Non-port-based activity, such as illegitimate root access, cannot be detected with this tool alone. We would like to incorporate other packet header fields (*e.g.*, from ICMP and IP) for non-port-based attacks, implement more information visualization methods (*e.g.*, zooming and mouseovers), and conduct human-computer interaction studies. ■

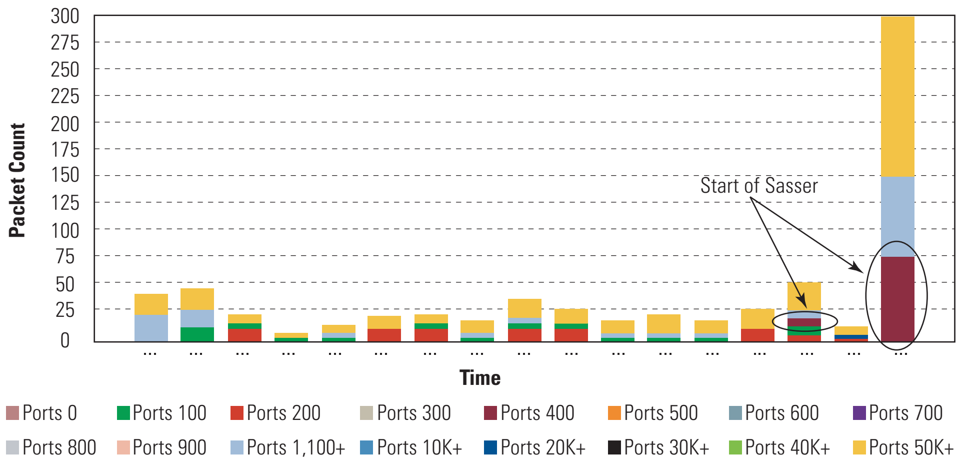


Figure 2. Incoming port counts for every 5 minutes during the Sasser worm attack.This capture is from the GT Honeynet.

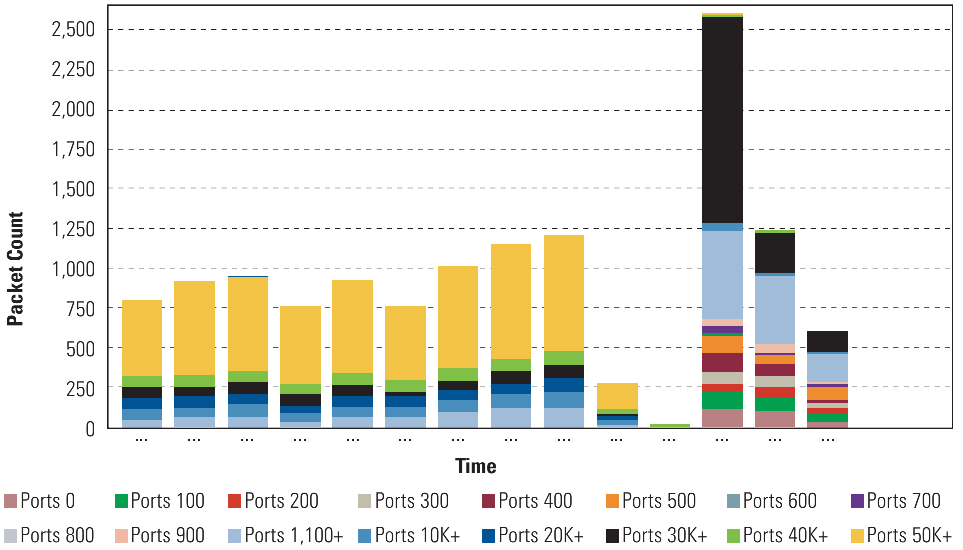


Figure 3. Slow network scan occurring over 30 minutes, plotted every two minutes.

References

[1] C. Cooper, "Snoozing About Security," vol. I. CNET Networks, 2005.

[2] P. Lyman, "How Much Information 2003?" October 2003.

[3] P. Jungck & S. Shim, "Issues In High-Speed Internet Security," *Computer*, vol. 37, pp. 36–42, 2004.

[4] R. Spence, *Information Visualization*. England: ACM Press, 2001.

[5] G. Combs, "Ethereal," Open Source, GPL.

[6] J. Levine, H. Owen, D. Contis, and B. Culver, "The Use of Honeynets to Detect Exploited Systems Across Large Enterprise Networks," Proceedings of the IEEE Workshop on Information Assurance, West Point, NY, 2003.

[7] "Honeynet project: Scan of the month," 2004.

[8] D. Keim, M. Hao & U. Dayal, "Hierarchical Pixel Bar Charts." *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, pp. 255–269, 2002.

[9] H. Koike & K. Ohno, "Snortview: Visualization System of Snort Logs," VizSEC/DMSEC'04, Washington DC, USA, 2004.

[10] K. Lakkaraju, W. Yurcik, A. Lee, R. Bearavolu, Y. Li, & X. Yin, "NVisionIP: NetFlow Visualizations of System State for Security Situational Awareness," VizSEC/DMSEC'04, Washington DC, USA, 2004.

[11] X. Yin, W. Yurcik, M. Treaster, Y. Li, and K. Lakkaraju " VisFlowConnect: NetFlow Visualizations of Link Relationships for Security Situational Awareness ," VizSEC/DMSEC'04, Washington, DC, USA 2004.

About the Authors

Ms. Kulsoom Abdullah | is a graduate research assistant at the Georgia Institute of Technology Communications Systems Center (*http://www.csc.gatech.edu/*). She is completing her PhD, and her research focuses are network security and visualization. Her research may be found at *http://users.ece.gatech.edu/~kulsoom/research.html*. She may be reached at kulsoom@gatech.edu.

Mr. Chris Lee | is a graduate research assistant at the Georgia Institute of Technology Communications Systems Center under Dr. John A. Copeland. He is completing his PhD and his current research focuses on security visualizations for ubiquitous deployment of security systems. His research may be found at *http://www.csc.gatech.edu/people/chrislee.html*. He may be reached at chris@ece.gatech.edu.

Mr. Gregory Conti | is an Assistant Professor of Computer Science at the US Military Academy, West Point, NY. He is currently at the Georgia Institute of Technology on a Department of Defense Fellowship where he is completing a PhD in Computer Science. His research may be found at *http://www.gregconti.com*.He may be reached at conti@cc.gatech.edu.

Dr. John Copeland | is the John H. Weitnauer, Jr. Chair at the Georgia Institute of Technology School of Electrical and Computer Engineering. In 2000, he co-founded Lancope, Inc., (*http://www.lancope.com*). His research interests include information visualization for computer security, network security and high-speed optical networks. Copeland has a BS, MS, and PhD in physics from the Georgia Institute of Technology. He is a Fellow of the IEEE, and received the Morris N. Liebmann award in 1970. He may be reached at john.copeland@ece.gatech.edu.

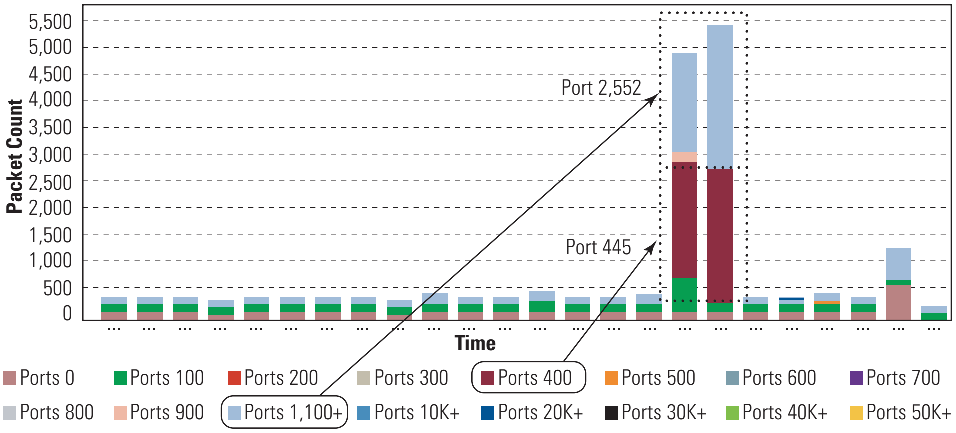


Figure 4. Sasser attack. This shows normally occurring probes and chatter. The spikes indicate a significant increase in the number of packets destined for two port ranges (the incoming on port 445 and the outgoing on port 2552).

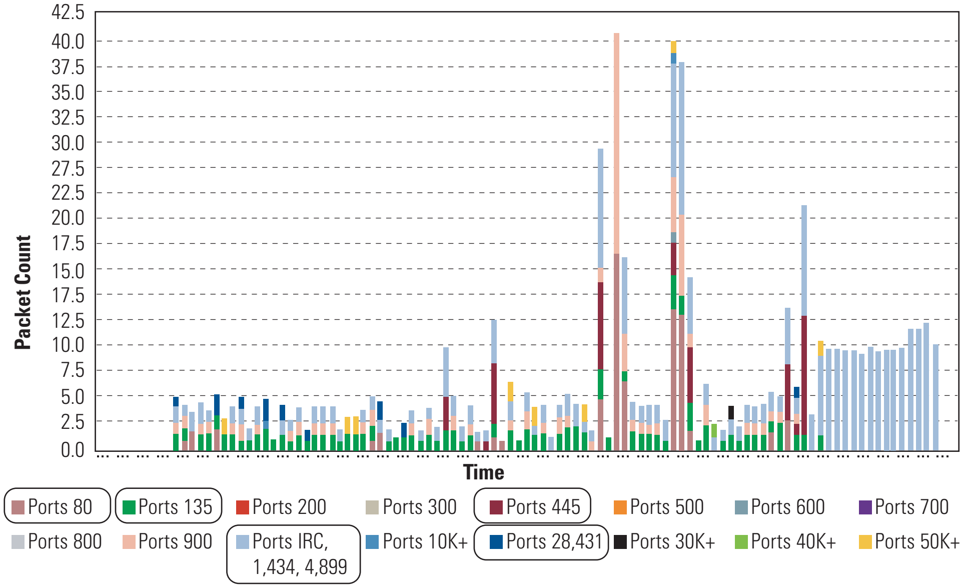


Figure 5. Stacked Histogram of Botnet Attack (Normalized)

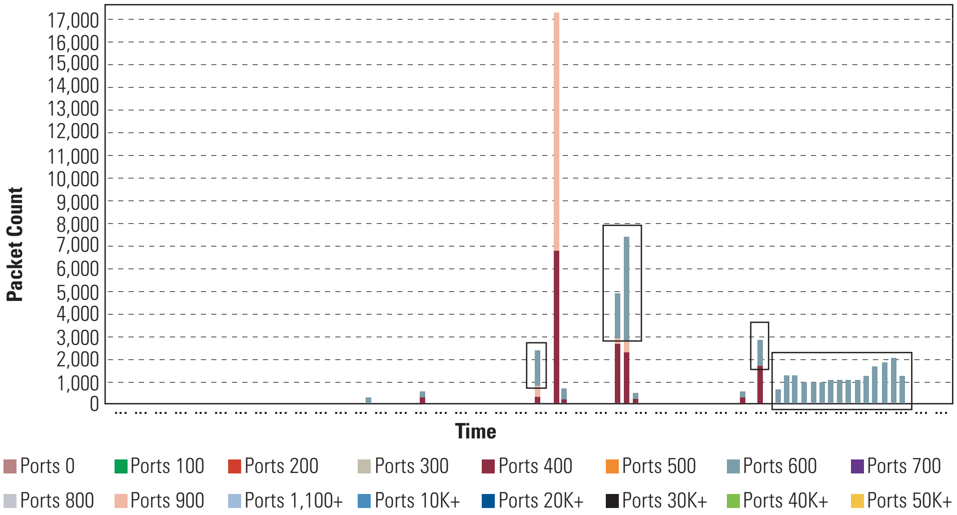


Figure 6. Botnet Traffic