Improving Security Visualization with Exposure Map Filtering

Mansour Alsaleh, David Barrera, P.C. van Oorschot *
School of Computer Science, Carleton University, Canada
{malsaleh,dbarrera,paulv}@scs.carleton.ca

Abstract

Graphical analysis of network traffic flows helps security analysts detect patterns or behaviors that would not be obvious in a text-based environment. The growing volume of network data generated and captured makes it increasingly difficult to detect increasingly sophisticated reconnaissance and stealthy network attacks. We propose a network flow filtering mechanism that leverages the exposure maps technique of Whyte et al. (2007), reducing the traffic for the visualization process according to the network services being offered. This allows focus to be limited to selected subsets of the network traffic, for example what might be categorized (correctly or otherwise) as the unexpected or potentially malicious portion. In particular, we use this technique to filter out traffic from sources that have not gained knowledge from the network in question. We evaluate the benefits of our technique on different visualizations of network flows. Our analysis shows a significant decrease in the volume of network traffic that is to be visualized, resulting in visible patterns and insights not previously apparent.

1 Introduction

Network security event monitoring is a time consuming and complicated process. Network security analysts are overwhelmed by massive amounts of audit log data that ideally would be analyzed for possible threats or malicious behavior. Different network-based and host-based security applications generate different types of textual logs. A log entry may indicate a prevention action taken by the monitoring application (e.g. blocking an attempt to access a closed port by a firewall) or an alert of possible malicious behavior (e.g. a worm signature detected by an intrusion detection system). Although some advanced network security tools can provide high-level overviews and reports, network security analysts often need to check the detailed logs in order to investigate a specific intrusion event. This time consuming process may fail to notice potential security threats. Difficulty in correlating different events within one or more log files and intentionally omitting from analysis log files thought to be less likely to carry important information add to the problem.

The use of visualization with network security data has continued to gain interest. Visual representation of network data, as opposed to textual representation, can help in analyzing a vast amount of data more quickly [4]. It takes humans much less time to recognize specific information or patterns in a picture than to detect the same in text. Humans are faster than computers in identifying some complex patterns and objects [17], and are able to identify new patterns never seen before. Most existing visualization tools provide a variety of representations of raw network data. In visual representations for mid- and large-size networks, the massive volume of network data makes it difficult to understand (to mine for useful information) and usually further processing of the textual data itself is still required.

Network scanning or reconnaissance is a common initial step in network intrusion attempts for identifying active hosts/ports on a network. The network exposure maps technique of Whyte et al. [22] records scan events that can be analyzed further to detect sophisticated scanning activities. A table is built of the services offered by a local network based on how internal hosts respond to incoming connection attempts, and inferences about probing remote hosts can be made based on whether probed services are actually offered. For example, external hosts that probe both any closed port on a local network machine and any open port are given special attention. The visualization technique presented in the present paper uses network exposure maps to help filter raw network data, in order to focus visualization efforts on data whose preliminary classification is as unknown or malicious traffic. This reduces the volume of traffic to be investigated for possible malicious behavior. Consequently, applying simple visualization techniques on the network traffic remaining after filtering yields much cleaner views, simplified by the removal of hopefully irrelevant data

*Version: Sept. 9, 2008. Authors listed in alphabetical order. The third author acknowledges NSERC funding under a Discovery Grant and as Canada Research Chair in Network and Software Security. Partial funding from NSERC ISSNet is also acknowledged.

and noise. It may also help analysts better correlate malicious events and discover coordinated external hosts.

We improve existing visualization methods in the filtering phase by speeding up the process of visually detecting patterns suggesting malicious traffic, obtaining valuable information from even simple visualizations and requiring less computing power and storage requirements to process or plot netflow data.

Organization. The sequel is structured as follows. Section 2 reviews exposure maps, and describes our dataset and network environment settings. Section 3 presents our filtering and visualization methodology. Section 4 illustrates our technique through case scenarios. Section 5 provides further discussion including limitations. We cover related work in Section 6. Section 7 concludes.

2 Background and Dataset

2.1 Review of Exposure Maps

Exposure maps [22] were proposed in part to reduce the computing resources necessary to detect sophisticated scanning. Instead of trying to detect scans through signatures, exposure maps track which ports are actively responding to outside connections. For a given internal host, all {IP address, port} pairs which respond to outside connection attempts are recorded. Collecting this information over all hosts in a local network makes up the *Network Exposure Map (NEM)*. The NEM is built over a training period: outgoing TCP flows containing SYN-ACK flags are observed and recorded,[1] with every host that was seen responding with SYN-ACK flags added to the NEM. Finally, the NEM is *vetted* (the offered services, as indicated by the NEM, are confirmed to be allowed by the network security policy). Ideally in the exposure maps technique the training period should be long enough to include legitimate traffic going to all open ports on the network in order to populate the NEM. Probes to closed ports during this training period will not establish sessions and therefore will not add entries to the NEM. Thus the training period does not need to be free from probes. In the production phase, each new incoming connection attempt is checked to see if it matches an entry in the NEM. If it does, it is labeled as legitimate traffic, otherwise as an atomic scan event. Memory (primary) for the exposure map itself is minimal, increasing linearly with the number of services offered on the network; memory (typically secondary) for the recorded scan events increases linearly with scanning.

[1] UDP ports in the NEM are added when 2 hosts communicating with the same source and destination port pairs (Host1 using *a* as a source port and *b* as a destination port, and Host2 using *b* as source and *a* as destination) are tracked within a small time period.

A malware infection can cause ports to transition from a closed (non-service) state to open. Such *trans-darkports* raise an alert and are added to the NEM as they transition, since these ports are responding to outside probes. An administrator tracks each trans-darkport to either shut down the port or verify that it conforms with the network security policy (e.g. a new authorized service was rolled out).

Whyte et al. [22] propose exposure maps for both sophisticated scan detection and automated response. We build on the exposure map idea of only caring about adversaries who have gained information from the local network (by discovering active services) and apply this concept to visualization. This allows us to significantly reduce the amount of information displayed in the visualization tool. Having a less cluttered visual display of netflow data helps the administrator detect low and slow scans as well as other patterns and stealthy attacks which might go unnoticed under traditional visualization techniques.

2.2 Exposure Map Generation

The dataset we use in this paper for our visualization experiments consists of a 28-day[2] PCAP [16] traffic capture on a university class C network with 62 Internet-addressable hosts. Only the first quarter of the class C has been assigned, leaving a darknet of 192 addresses. The network sniffer used to capture the dataset was placed on the external interface of the border firewall so traffic between internal hosts was not captured.

Netflow data was generated from the packet capture using the Argus suite [1]. Each flow was then entered in a MySQL database. Each database entry contains standard netflow fields (start and end time, source and destination ports, protocol, source and destination IPs, session flags, byte size and packet count) along with unique identifiers for each flow. Also, each of the netflow fields was stored in a column and indexes were generated for the columns storing frequently used values (source IP, source port, destination IP, destination port) to minimize query response time as in Stockinger et al. [20].

Host	Port	Protocol
11	25 (SMTP), 631 (IPP), 993 (IMAPS)	TCP
11	53 (DNS)	UDP
13	22 (SSH), 80 (HTTP), 443 (HTTPS)	TCP
13	53 (DNS)	UDP
58	22 (SSH)	TCP

Table 1. Network Exposure Map (NEM)

[2] The capture began on Nov. 12, 2007 and ended on Dec. 9, 2007, with an 8 hour gap starting at 3 p.m. on Nov. 25, 2007 due to a power outage.

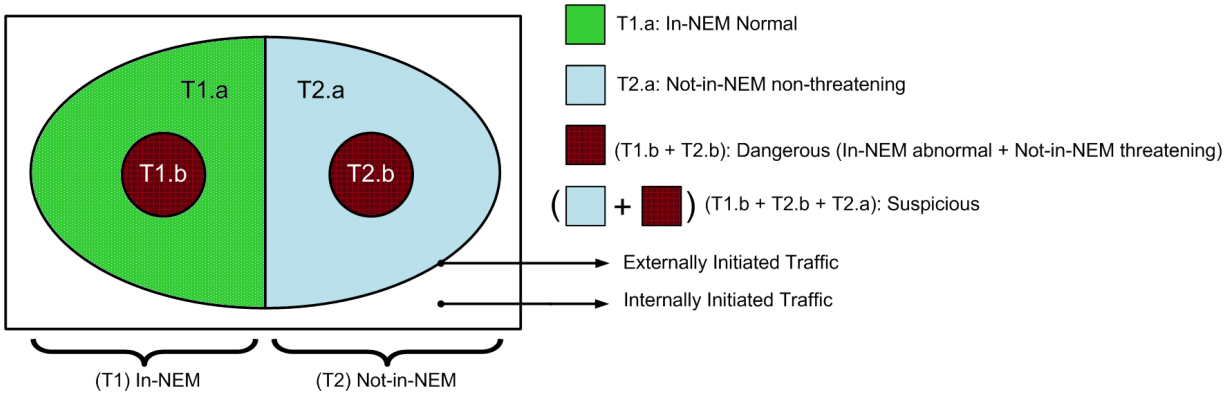


Figure 1. Graphical representation of filtered flow subsets

Table Name	*no. of flows*	*no. of distinct source IPs*	*no. of packets*	*total size of all flows*
Flows (all externally initiated traffic)	863,430	13,442	15,941,933	10,051 MB
T1.b (In-NEM abnormal)	690,660	3,816	1,361,669	89 MB
T2 (Not-in-NEM)	23,683	78	565,148	109 MB
T3 (Suspicious)	714,343	3,816	1,926,817	198 MB
T4 (Dangerous)	39,950	78	597,648	111 MB

Table 2. Table Statistics. MB denotes megabytes.

To generate the exposure map for this dataset, we query the database for any flows (with a source IP within the local network) that responded with a SYN-ACK packet to incoming connection attempts. These flows include local IP addresses and ports that are actively responding to incoming TCP requests. For UDP, we query the database for hosts communicating with the same source and destination port pairs. We build the NEM accordingly and verify that it conforms with the network security policy. The resulting NEM is shown in Table 1.

3 Methodology of Exposure Map Filtering

This section explains our process of filtering and visualizing the netflow data. Externally initiated flows from the dataset are categorized into a number of disjoint sets (see Fig. 1), in logical tables with semantics as follows.

Table T1: In-NEM. This table contains flows destined to a host/port combination offering an authorized service (i.e., to an authorized open port in the local network). This table is also logically partitioned into two sub-tables.

T1.a: In-NEM normal. This table contains flows that are considered ordinary, since their source IP addresses have only attempted connections to authorized services offered by the network in question (i.e., destined to an authorized open port).

T1.b: In-NEM abnormal. This table contains flows initiated by source IP addresses that also have flows in T2. We label these flows 'suspicious' because normally, a host does not attempt connections to closed ports while also accessing legitimate services.

Table T2: Not-in-NEM. This table contains flows destined to a host/port combination for which no authorized service is offered (i.e., closed port). It is logically partitioned into two sub-tables.

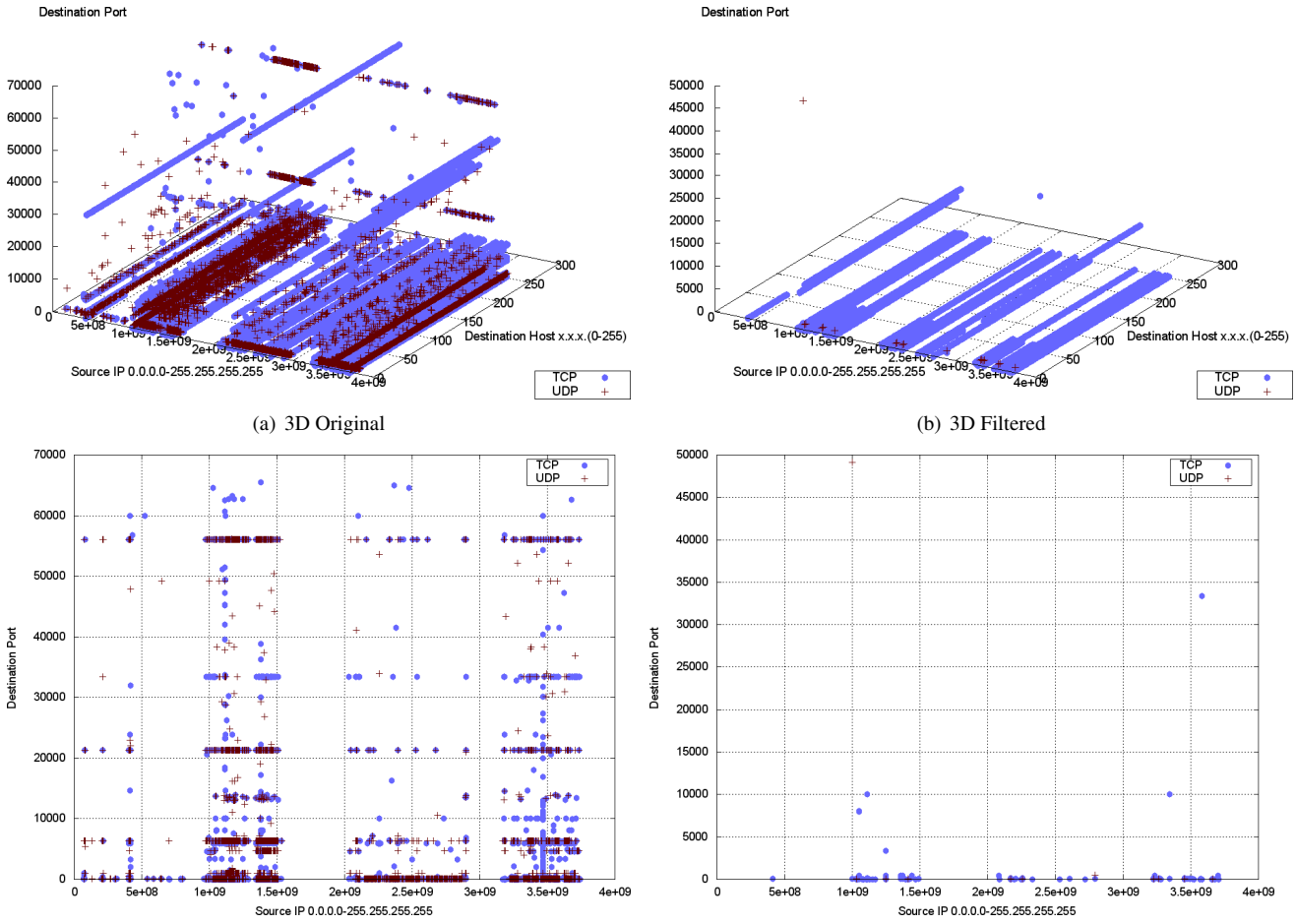
T2.a: Not-in-NEM non-threatening. This table contains flows in T2 and whose source IP addresses have no flows in T1. Exposure map filtering assumes these connection attempts are not a significant threat to the target network since sources, all of whose probes have been to closed ports, have not learned what is considered significant information from the target network (i.e., have not learned what services are offered).

T2.b: Not-in-NEM threatening. This table contains flows in T2 and whose source IP addresses also have flows in T1. Thus, the source IP address of these flows have queried both legitimate offered services and closed ports.

Table T3: Suspicious. This table includes all flows in T2 (T2.a and T2.b) plus T1.b. We call this 'suspicious traffic' because these source IP addresses have probed at least one closed port in the network.

Table T4: Dangerous. This table includes all flows in T1.b plus T2.b. This represents traffic from IP sources that probed at least one closed port and also attempted to connect to an open port. According to the philosophy motivating the exposure maps technique, these are more likely to represent malicious flows since these IP sources, if adversaries, might attempt to send exploits to the open ports that they have discovered.

The full dataset of externally initiated traffic described in Section 2.2 is stored in a data structure called the **flows** table. In addition to this, the following subsets of the previously described logical tables are actually built: tables T1.b, T2, T3, and T4 (and rather than duplicating data, only links to the flows table entries are stored in these tables). Statistics for each of these tables, for the dataset described in Section 2.2, are presented in Table 2.



(a) 3D Original

(b) 3D Filtered

(c) 2D Original

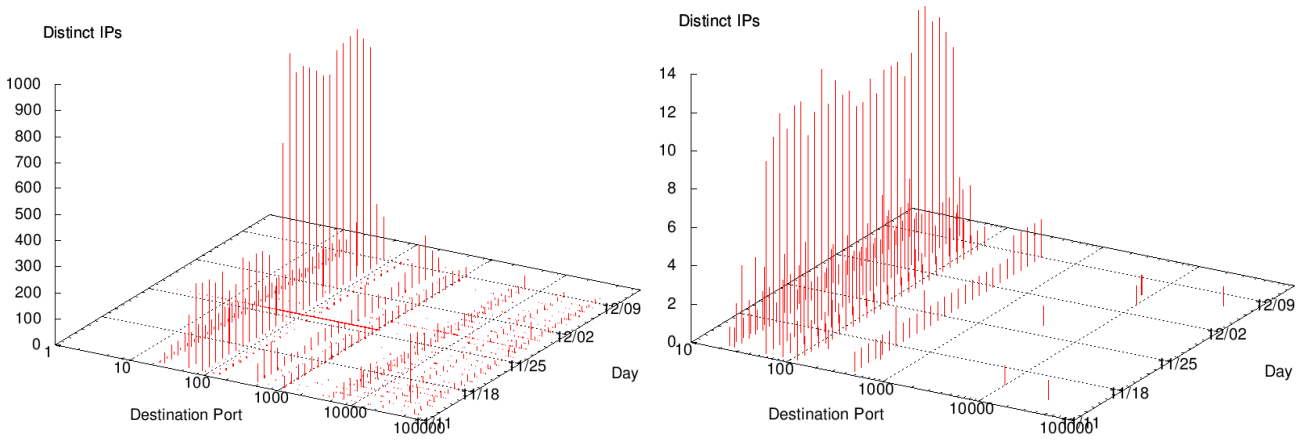
(d) 2D Filtered

Figure 2. Destination IP and port from full source IP address (best viewed in color)

4 Illustrative Visualizations

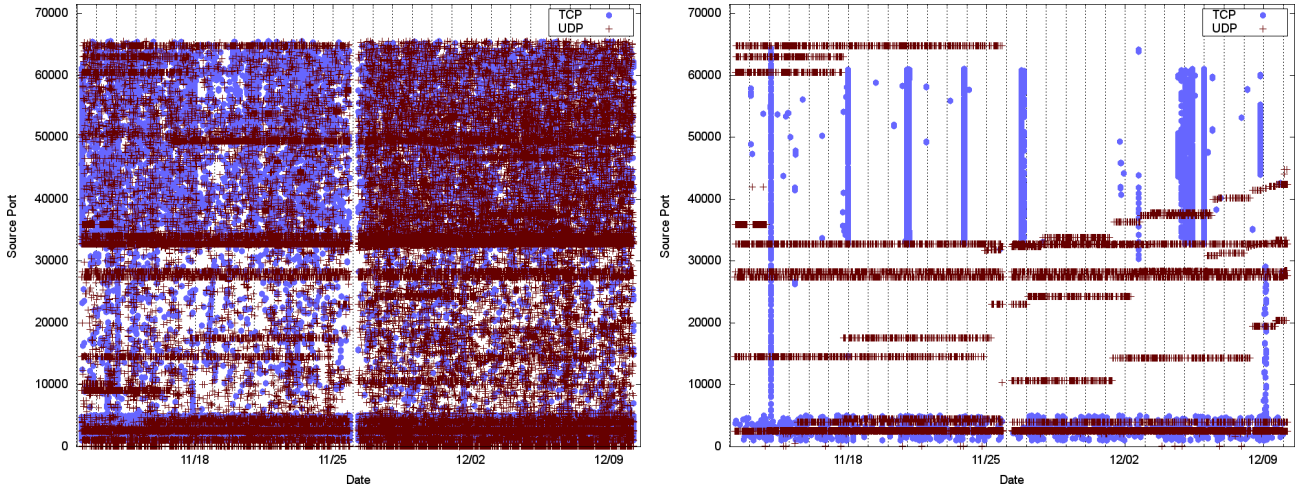
Using the filtering as indicated by the logical tables in Section 3, we proceed to plot netflow data on either a 2D or 3D space. Although there are a large number of different types of graphs (i.e. choices of data features to plot on the *x, y* and *z* axes), in this section we have selected 9 sets of graphs to illustrate the advantages of the filtering technique. The majority of these are simple or known graph types. Each set of graphs is intended to contrast the information conveyed by the visualization before and after filtering. In all cases, the patterns in malicious activity were discovered through this filtered visualization process itself, with valuable insight gained from the filtered visualizations over the original unfiltered graphs. For each of the examples in this section, in the unfiltered visualization we have plotted all externally initiated flows as item (a). In the filtered visualization (item (b) in each graph pair) we have only plotted flow data from the *dangerous table* (T4) of Section 3. We emphasize that in practice, we expect that the analyst will only need to study each item (b), with little or no information gained from comparing (a) with (b).

Figure 2 graphs the full source IP address (plotted as an integer from 0 to approximately 4.2 billion), the target destination host and the destination port. Figure 2(a) shows a high number of source IPs probing a single port on the entire class C destination network and dense areas around low-order ports. On Figures 2(c) and 2(d), the original 3D visualization is projected to a 2D view showing the exact destination ports more clearly, while hiding the destination IP address. Figures 2(a) and 2(b) are similar to the "Spinning Cube of Potential Doom" [9], except that we plot network flow data as opposed to Intrusion Detection System (IDS) logs. Due to the large number of horizontal scans (probing a single port on all destination IP addresses as noted by bottom-left to top-right diagonal lines) displayed, a security analyst might have trouble identifying which scans warrant further analysis from Figure 2(a). However, most (if not all) horizontal scans in Figure 2(b) likely reveal some type of malicious activity. For example we notice some horizontal scans that only target the first 64 addresses of our subnet, which suggest that the scanner is aware of our network topology. The data that was automatically removed in moving from Figure 2(a) to 2(b) was, as previously noted, classified as non-threatening by the exposure map filtering. For example, the left-most horizontal scan on Figure 2(a) belongs a single source probing all hosts on the destination network for port 32000 TCP which is not offered. On Figure 2(b) we randomly selected a scan in the middle



(a) Original (b) Filtered

Figure 3. Number of distinct source IP addresses per destination port



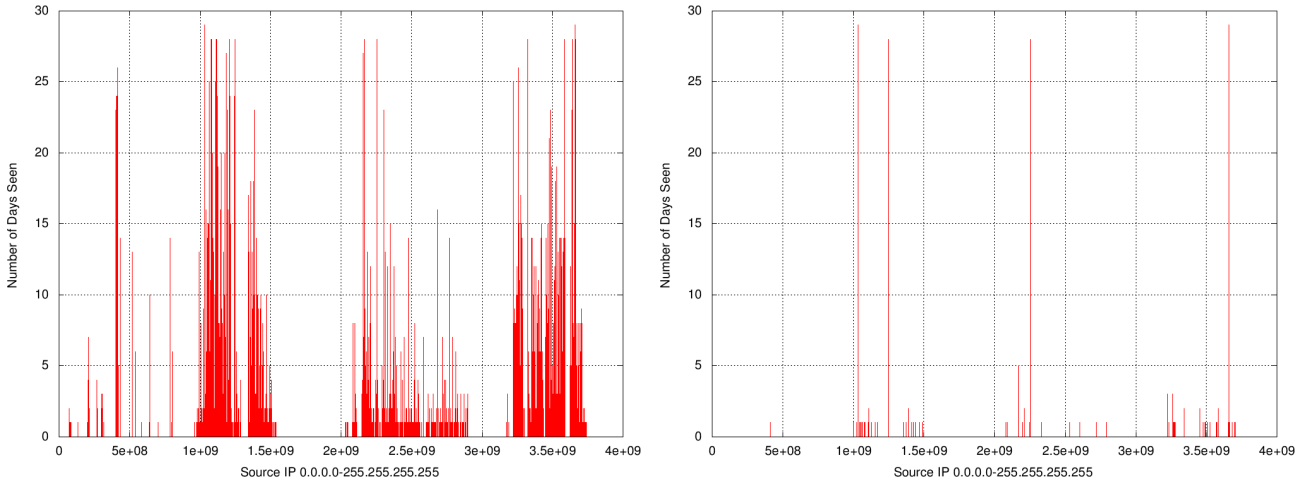
(a) Original (b) Filtered

Figure 4. Source port over time (best viewed in color)

of the graph (selection not shown). After zooming in, we found the port targeted was 9999 so we searched the database for the IP addresses that probed this port, and found a single source that tried to connect to ports 143 TCP (Internet Message Access Protocol) and 9999 TCP on all hosts in this destination class C. That same source also attempted to connect to port 25 on all hosts in the destination subnet, finding the SMTP server along the way. This source may be categorized as an adversary, having learned something from the target network and might be more closely monitored for future intrusion attempts. On Figure 2(c), we notice a single source IP address attempting connections to a large number of TCP destination ports below 40000 (vertical line on the right). Querying the database suggests this traffic is unimportant because the source is trying a large number of destination ports on a non-existing host; note this traffic is absent from Figure 2(d).

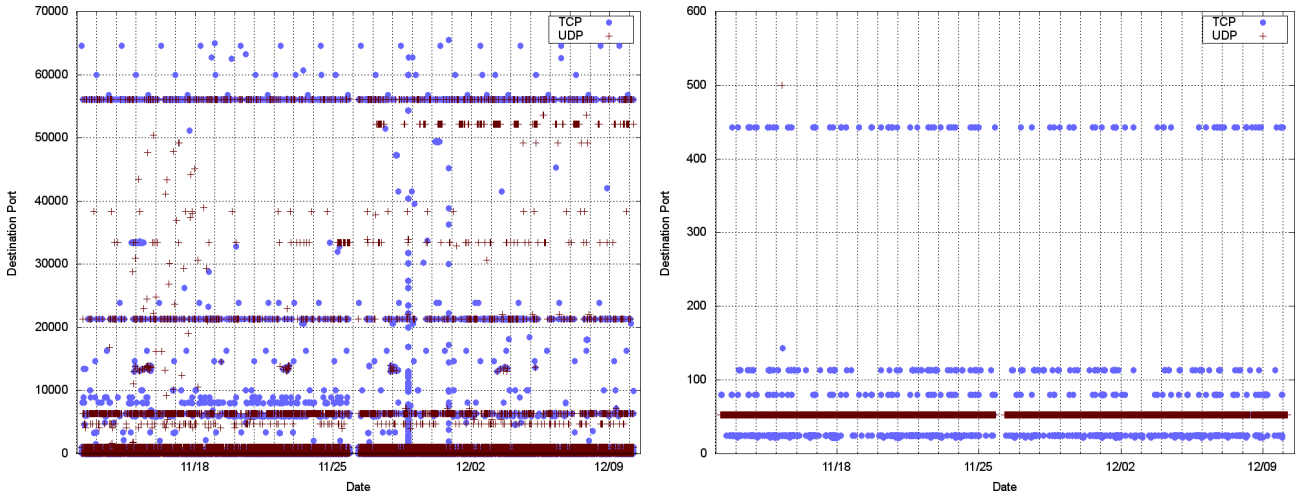
Figure 3 presents a count of the number of unique sources that were seen targeting each destination port on a particular day and illustrates a high number of sources attempting connections to ports 1-1000. This type of graph can help detect either groups of collaborating hosts or those infected by the same worm. In Figure 3(a), we notice an increase in inbound flows to port 53 beginning on December 1. Further investigation by means of database queries showed that the vast majority of sources going to port 53 on the target network's DNS server are not attempting connections to any other ports/hosts, indicating that the DNS queries are probably legitimate. Upon viewing the filtered graph in Figure 3(b) (which has been autoscaled due to less volume), we see a consistently high number of sources targeting port 53 across the entire capture period. Motivated by this visual cue, we discovered (by reviewing text flow records) that these are 11 unique sources probing port 53 both TCP and UDP every day.

Figure 4 plots the source port used for incoming flows over the entire capture period. Source ports might give insight as to what operating system is being used, or what type of scan is being performed. Source port distribution is also useful for detecting the spread of worms [21]. Source ports are usually allocated randomly which explains the high degree of clutter in Figure 4(a). Figure 4(b) on the other hand shows a cleaner view of the source



(a) Original (b) Filtered

Figure 5. Frequent originators among source IP addresses

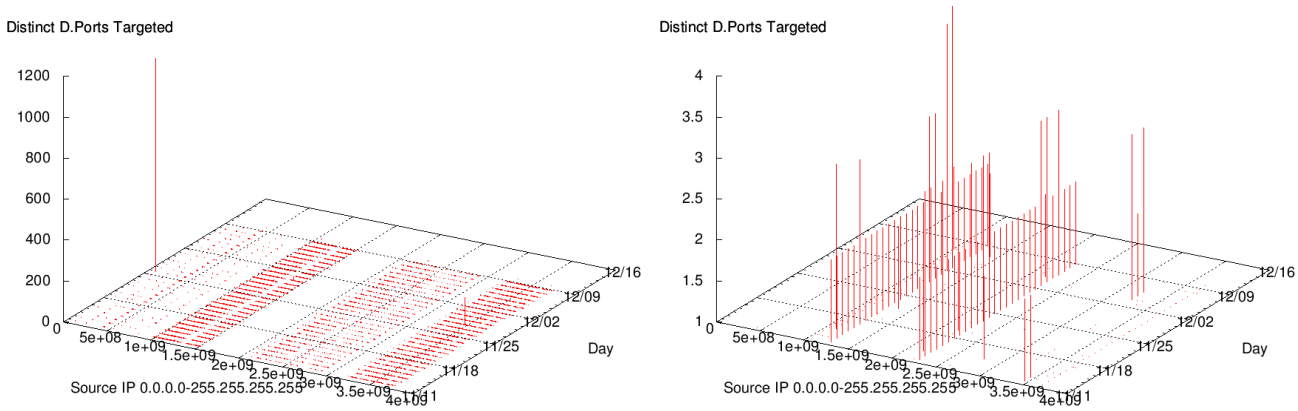


(a) Original (b) Filtered and zoomed-in

Figure 6. Destination port over time (best viewed in color)

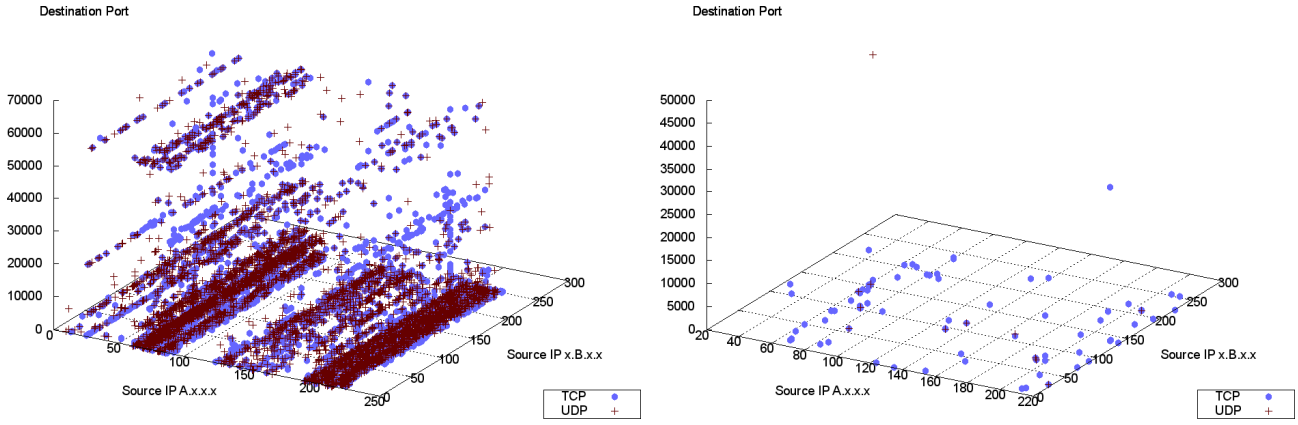
ports being used by what we call dangerous sources. Though there are quite a few patterns, we focus our attention to the 4 evenly spaced vertical lines in the center of the plot. Triggered by this visual cue, we queried the database for the sources that match those source ports, and we found that 4 distinct IP addresses in separate class A networks were attempting to bruteforce secure shell (SSH) accounts. Each one of the 4 sources was found to be scanning the full destination class C for any hosts which offer the SSH service (TCP port 22), and upon finding exactly the 3 hosts which responded (note the 3 hosts serving the SSH service in Table 1), changed their bruteforce attack to focus only on those hosts.

Figure 5 gives the number of days a specific source IP address probed the network. IP addresses that probe the network repeatedly might be considered for additional analysis. Note that in Figure 5(a), a large number of source IP addresses attempt connections over 15 times in the 4 week capture. The filtered view leaves only only 4 peaks, representing 4 groups of sources each probing during more than 25 days. This view is useful, as it leaves the analyst with far less information to analyze; what the filtering technique automatically removed was legitimate traffic as well as probes from source IPs going to only closed ports. By querying the database for the source IPs corresponding to each one of these 4 peaks, we were able to find two small networks, with addresses allocated to Chinese ISPs, in which hosts exhibited the exact same behavior. One network of 6 computers probed ports 53 TCP and UDP (Domain Name Service) on 2 of the destination network servers roughly 10 times per hour during the entire capture period. The other network of 4 computers probed port 53 TCP and UDP as well on an hourly basis but only on one of the target servers. These 10 IP addresses should be monitored closely, as they have all found services the network is offering and are exhibiting reconnaissance behavior. As another interesting observation in Figure 5(b) we see a noteworthy division between line heights. We see either sources that return almost every day, or



(a) Original (b) Filtered

Figure 7. Distinct destination ports targeted



(a) Original (b) Filtered

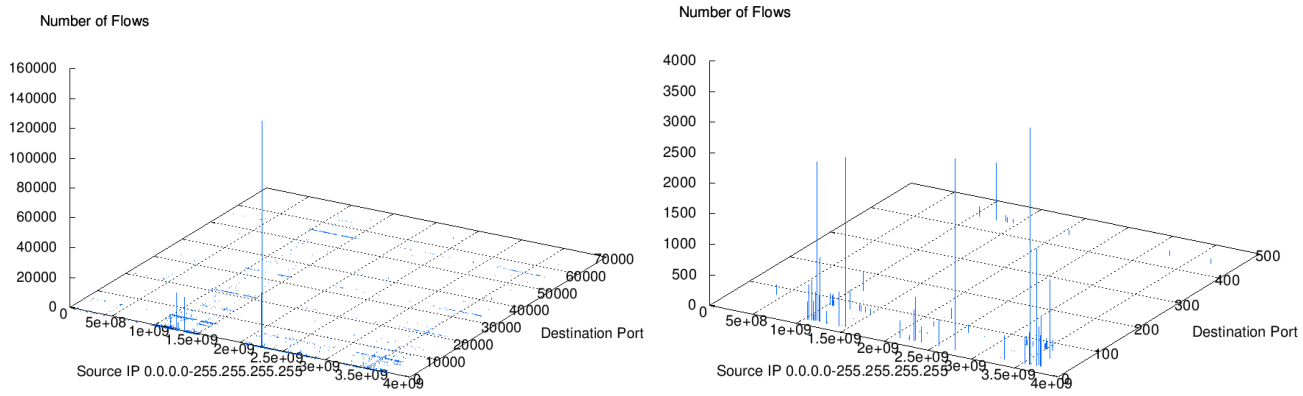
Figure 8. Destination port from class B source networks (best viewed in color)

sources that come back less than five days. This suggests potential coordinated activity among sources with the same behavior.

Figure 6 shows the local destination port targeted over the entire capture period. Some ports are frequently targeted, particularly low-order ports. On November 28, we see a scan taking place (noted by the vertical line on Figure 6(a)). Although full vertical scans are less common today, this choice of visualization (independent of exposure map filtering) prominently highlights remote hosts performing such scans. A zoom-in version of the filtered graph is displayed in Figure 6(b) and shows that a small number of low-order ports are probed throughout the capture period. An interesting observation is that the vertical scan from November 28 is removed by the filtering. Upon further investigation through database queries, we found that the source of the scan in question probed 1036 ports on a non-existing host on the destination network. All of these probes went unanswered, and the source did not try to connect to any other hosts. Had it tried to connect to other target network hosts and found an open port, the flows from that source IP address would have been categorized differently through the exposure map-based methodology of Section 3, and not removed by the automatic filtering. These conditions result in the filtering system (apparently safely) ignoring traffic generated by this host, and omitting it in Figure 6(b).

Figure 7 shows the number of distinct destination ports targeted for each source IP on a specific day. A high number of distinct destination ports coming from a single source is what we expect to attract an analyst's attention since this would potentially imply malicious activity. Notice that in Figure 7(a), a single source attempting connections to nearly 1200 ports ruins the visualization in its original form, likely encouraging an analyst to switch to a logarithmic scale. However, in Figure 7(b), the large spike was part of the data automatically filtered out, and the remaining lines reveal interesting information. In this particular example, every source classified by exposure maps as dangerous visited at most 4 distinct destination ports. Cued by this new view we queried the database for any hosts that attempted connections to any 4 distinct destination ports, and found 2 sources probing for ports 80, 443, 8000, and 8080 (commonly used web server ports) on the first 64 addresses of the target network.

Figure 8 shows in a 3D plot the targeted ports on the destination network probed from class B external networks. The x and y axes represent the first and second octet of the source IP address respectively. Each point represents a single inbound flow. For example note that for a source IP of 200.200.x.x, there are several



(a) Original (b) Filtered

Figure 9. Number of flows per source IP address and target port

destination ports targeted (vertically). This type of plot is useful for pinpointing the class B subnet sources of incoming traffic and the most targeted destination ports for each one. We notice 2 sparse vertical lines in Figure 8(a) which suggest that several hosts in the same class B network are attempting connections to a large number of destination ports on the target network. The lines disappear after filtering. The filtering approach takes the position that those sources did not learn anything useful from (i.e., got no responses from) the target network, and therefore (apparently being) not a threat, can be safely filtered out. Figure 8(b) shows a smaller number of plotted flows, allowing the analyst to notice that the distribution of dangerous class B's appears mostly random, and that nearly all hosts in each network were targeting low-order ports.

Figure 9 displays the traffic volume (in number of flows) at each destination port along with the source of this traffic (plotted on the x axis). The noticeable vertical line in the front of Figure 9(a) represents a high volume of traffic coming from a small number of source addresses to a few destination ports. This type of plot is helpful in highlighting source IPs sending high traffic to specific network ports. Similar numbers of flows going to different destination ports might indicate some correlations between the sources. Although this type of activity wasn't captured in Figure 9, it was present in Figures 5(b) and 7(b) and helped detect hosts that appear to be working together. The spike in Figure 9(a) turns out to come from the same parent Class B network as the dataset's local network. When we queried the database for the flows that make up that spike, we discovered a single source IP attempting connections to port 445 (Microsoft Windows Shares) on all 254 destination IPs on the target network. This spike is absent in the filtered graph (Figure 9(b)) because this service is not offered on any of the target network's hosts or servers. Since the target network won't reply to this type of activity, and the probing source IP address did not probe any open ports, exposure map filtering reasons that these connection attempts are not an actual threat to the target network. Thus omitting this activity from the displayed data (as in Figure 9(b)) simplifies the interpretation of the remaining network activity, with no loss – the omitted activity is not among the data in need of review by the analyst. One could argue, however, that the local network analyst does have a vested interest in knowing that probing machine's address and behavior, since in this case it is a local machine which the analyst might be able to shut down. Our filtering tools could be easily modified to special-case machines from the local network (or other specified classes of machines) and not filter out suspicious traffic originating from them. The visual patterns in the sparser activity that remains cued us to explore the database records related to the peaks in Figure 9(b). We found that these correspond to SSH bruteforce attacks.

5 Further Discussion

We emphasize the noticeable reduction in traffic plotted from the original to the filtered view in all of the visualizations presented in Section 4. Exposure map-filtered views in all cases provided a fast way to focus on what we have called dangerous traffic. While we do not claim that this dangerous traffic is impossible to discover in the original (unfiltered) views, the filtered views did help us, and we believe will help the general analyst speed up network flow analysis in many cases. Section 4 presented a limited number of examples of how our filtering technique may help security analysts better understand malicious activity in their network.

While our filtering technique proved to be useful in many scenarios, there are of course cases where attackers might go unnoticed. Our proposal is based on exposure maps, and therefore suffers from inherent limitations of exposure maps such as the assumption that attackers, while trying to gain knowledge about the target network, will likely attempt connections to closed ports only, or both listening (open) *and* closed ports. Therefore, flows originating from an attacker's IP address lucky (or clever) enough to probe only services the target network is offering will tend to look like legitimate connections and not get visualized. An adversary with access to a large botnet could perform a scan with a subset of those machines, and subsequently launch an attack on only identified open ports with hosts that were not used during the scan. The exposure maps technique would classify all the scanning IP addresses as suspicious or malicious (except addresses which only scanned open ports), but the new activity of the new machines would be classified normal. In a more advanced scenario, each zombie machine could be used to probe only a single port on a single host, and then attacks on the open ports could be launched with new zombies. Such an attack would appear to be

difficult to detect, although a visual pattern of a coordinated scan might be detected. As another limitation, large volumes of possibly malicious traffic targeting the network in question might lead to cluttered filtered visualizations. This could be mitigated by using further basic filtering such as limiting the filtered visualization to specific destination hosts, ports, or protocols.

Many of the potential problems noted above are not specific to our particular visualization proposal. Conti et al. [5] discussed a variety of possible information visualization attacks targeting either the users of visualization tools or the software and hardware of these tools. We believe that essentially any network analysis visualization tool or approach will have its own drawbacks, and in many cases a combination of approaches may be required. We believe that our filtering approach remains useful even though the exposure maps philosophy may result in mis-classifying some traffic (e.g. shifting focus away from specific IP addresses, when in fact those might actually be dangerous or malicious).

Given a massive amount of raw network traffic and logs, it is not easy for security analysts to monitor network activities in real-time using most existing visualization techniques. We believe our technique could be easily adapted to the process of filtering network flows on-the-fly with immediate visualization, allowing analysts to focus exclusively on visualizations of suspicious or dangerous network flows in a real-time fashion. For each new externally initiated flow, there are at most two comparisons necessary to decide whether the new flow should be added to the dangerous table (T4 in Section 3). First, we use the NEM to check if the flow is destined to a host/port combination for which no authorized service is offered. If so (i.e., it is not-in-NEM) then the flow is added to the dangerous table (the table that is visualized in part (b) of all of our Section 4 examples) only if the flow source IP address has also previously attempted to connect to an open port in the network (for example, within a predefined time window). Otherwise, if the flow is attempting to connect to an open port, the flow will be added to the dangerous table only if the flow source IP address has probed at least one closed port in the network. Note this can be naively facilitated by using the equivalent of two bit array data structures, for each source IP address seen to date (or all 2^{32} IPv4 addresses in the extreme case), tracking if that source has yet probed (a) an open port, and (b) a closed port. The processing time for each network flow thus has a small constant upper bound.

6 Related Work

Interest in information security visualization has grown rapidly in recent years, providing network security analysts with new tools and methods for visually identifying and classifying network attacks. Most of these visualization systems to date are based upon the information visualization mantra: *overview first, zoom and filter, details on demand* [19]. In practice, these systems first display large amounts of data, and then allow the analyst to zoom and filter into areas of the visualized data that might be of interest. Looking for patterns and clusters in graphs is an obvious approach to detect attacks; however, modern attacks are getting more sophisticated and stealthier, making it difficult for the analyst to select the appropriate areas to zoom into. Often, this situation forces the analyst into an inefficient repetitive process of zooming in and viewing details, only to realize that the area zoomed into is not of interest. One of the advantages of visualization should be to speed up data interpretation by leveraging human visual processing power. However, today's highly distributed attacks and excessive Internet background radiation [15] make it difficult to use standard visualization tools to their full potential. Conti et al. [4, 6] discuss visualization systems which filter data but leave it up to the analyst to select which records to display. The prefiltering phase involves removing unwanted fields from flows or formatting the data; it does not automatically remove any full flows.

The Portvis tool [11, 2] allows the analyst to see port activity on a range of hosts over a defined period of time, offering a very high-level method of detecting malicious activity. Attackers may be able to bypass detection by using frequently used ports, or a small number of packets. The NvisionIP tool [23] uses a grid-based visualization to plot the source IP address, and colors to identify ports accessed by this address. This type of visualization can offer interesting insight, but may be prone to clutter when a large number of sources target a large number of ports. Honeynets are commonly used to analyze malicious network behavior. Grizzard et al. [8] compared the Georgia Tech honeynet data to real-world data submitted by volunteers. Using histograms over large time scales, they were able to visualize large amounts of data. While able to detect large spikes in activity related to large-traffic worms such as Blaster, stealth scans are largely undetectable by these histograms.

An alternate approach is to characterize or model malicious behavior through visualization. Muelder et al. [13] suggested a tighter interaction between the typical overview and detail phases. Conti et al. [3] explored approaches to create a set of images that visually describe scanning tool behavior (such as NMAP [14]) under different operating systems. While this is useful for understanding adversaries who use popular tools, minor modifications to these tools can generate a different (and thus evasive) visual footprint. Muelder et al. [12] also pursue classification of suspicious network traffic, using associative memory with neural networks to reduce noise and identify scans. A tool for visualizing horizontal and vertical scans is the Spinning Cube of Potential Doom [9] which highlights different types of scans. Advanced techniques such as slow scans or highly distributed scans may be hidden by the noise of normal traffic and Internet background radiation. A similar concept for finding horizontal or vertical scans was also reviewed by Gates et al. [7] in a 2D space. In our research to date, we have found few scanners that scan enough ports/hosts to create horizontal or vertical lines in appropriate graphs. This might be attributed to attackers trying to avoid triggering alarms. Gates et al. used the concept of unique source IP addresses per hour to detect malicious activity. Finally, the Isis tool [18] allows visualization of an intrusion through variable time-scales.

7 Concluding Remarks

Our visualization proposal relies heavily on the exposure maps technique [22] for network traffic filtering. This significantly reduces the amount of network traffic displayed for security monitoring and analysis, with plots of the 'dangerous' flows focusing attention on traffic from sources that have probed at least one

closed *and* one open port. Although not a foolproof way of detecting all malicious traffic, we believe this approach proves useful to highlight both common and some advanced adversaries, and compromised hosts. In contrast, the majority of available tools and methodologies for visualization we are familiar with offer filtering features but require the analyst to correctly input appropriate filtering parameters. Incorrect parameters may result in processing overhead to display unnecessary network traffic or events, and analysts may then have trouble finding the needle of malicious activity in the haystack of network activity.

We believe the simple visualizations for security-related analysis of network traffic presented herein demonstrate the effectiveness of filtering out the "known good" (or assumed good) traffic, and the "harmless bad" traffic, allowing focus on a specific type of suspicious traffic. Although the dataset that we used was from a relatively small university network, even on this network the visualizations of the unfiltered flows cause difficulties in finding malicious activities due to the volume and richness of the traffic. While this dataset has proven effective for illustrating our approach, future work includes experimenting further with larger datsets from larger networks (e.g. perhaps [10]), and testing the effectiveness of combining our proposal with other existing visualization approaches [11, 18, 9, 12].

References

[1] Argus Suite. `http://qosient.com/argus`. Accessed January 2008.

[2] C. Muelder, K.-L. Ma, and T. Bartoletti. Interactive visualization for network and port scan detection. In *RAID'05: Proceedings of Recent Advances in Intrusion Detection*, September 2005.

[3] G. Conti and K. Abdullah. Passive visual fingerprinting of network attack tools. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, pages 45–54, New York, NY, USA, 2004.

[4] G. Conti, K. Abdullah, J. Grizzard, J. Stasko, J. A. Copeland, M. Ahamad, and C. Lee. Countering security analyst and network administrator overload through alert and packet visualization. *IEEE Computer Graphics & Applications*, 26(2):60–70, March/April 2006.

[5] G. Conti, M. Ahamad, and J. Stasko. Attacking information visualization system usability overloading and deceiving the human. In *Proceedings of the 2005 symposium on Usable privacy and security (SOUPS 05)*, pages 89–100, 2005.

[6] G. Conti, J. Grizzard, M. Ahamad, and H. Owen. Visual exploration of malicious binary objects using semantic zoom, interactive encoding and dynamic queries. In *Proceedings of the Workshop on Visualization for Computer Security (VizSEC 05)*, pages 83–90, October 2005.

[7] C. Gates, J. McNutt, J. Kadane, and M. Kellner. Detecting scans at the ISP level. Technical Report CMU/SEI-2006-TR-005, Software Engineering Institute, Carnegie Mellon University, 2006.

[8] J. Grizzard, J. Charles Simpson, S. Krasser, H. Owen, and G. Riley. Flow based observations from NETI@home and honeynet data. In *Proceedings of the Sixth IEEE Systems, Man and Cybernetics Information Assurance Workshop*, pages 244–251, June 2005.

[9] S. Lau. The spinning cube of potential doom. *Communications of the ACM*, 47(6):25–26, 2004.

[10] LBNL/ICSI Enterprise Tracing Project. `http://www.icir.org/enterprise-tracing/Overview.html`.

[11] J. McPherson, K.-L. Ma, P. Krystosk, T. Bartoletti, and M. Christensen. Portvis: a tool for port-based detection of security events. In *VizSEC/DMSEC '04: Proceedings of the ACM Workshop on Visualization and Data Mining for Computer Security*, pages 73–81, New York, NY, USA, 2004.

[12] C. Muelder, L. Chen, R. Thomason, K.-L. Ma, and T. Bartoletti. Intelligent classification and visualization of network scans. In *Proceedings of the Workshop on Visualization for Computer Security (VizSEC 07)*, October 2007.

[13] C. Muelder, K.-L. Ma, and T. Bartoletti. A visualization methodology for characterization of network scans. In *Proceedings of the Workshop on Visualization for Computer Security (VizSEC 05)*, pages 29–38, Washington, DC, USA, 2005. IEEE Computer Society.

[14] NMAP. `http://nmap.org`. Accessed April 2008.

[15] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson. Characteristics of internet background radiation. In *IMC '04: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, pages 27–40, New York, NY, USA, 2004.

[16] PCAP. `http://tcpdump.org`. Accessed January 2008.

[17] M. Peck. A brainy approach to image sorting. `http://www.spectrum.ieee.org/apr08/6121`, April 2008.

[18] D. Phan, J. Gerth, M. Lee, A. Paepcke, and T. Winograd. Visual analysis of network flow data with timelines and event plots. In *Proceedings of the Workshop on Visualization for Computer Security (VizSEC 07)*, 2007.

[19] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. Technical Report UMCP-CSD CS-TR-3665, College Park, Maryland, 1996.

[20] K. Stockinger, E. W. Bethel, S. Campbell, E. Dart, and K. Wu. Detecting distributed scans using high-performance query-driven visualization. In *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*, New York, NY, USA, 2006.

[21] A. Wagner and B. Plattner. Entropy based worm and anomaly detection in fast IP networks. *14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise*, pages 172–177, June 2005.

[22] D. Whyte, P. C. van Oorschot, and E. Kranakis. Tracking darkports for network defense. *Twenty-Third Annual Computer Security Applications Conference, ACSAC 2007*, pages 161–171.

[23] W. Yurcik. Tool update: NVisionIP improvements (difference view, sparklines, and shapes). In *Proceedings of the Workshop on Visualization for Computer Security (VizSEC 06)*, New York, NY, USA, 2006.