IDGraphs: Intrusion Detection and Analysis Using Stream Compositing

Pin Ren, Yan Gao, Zhichun Li, and Yan Chen
Northwestern University

Benjamin Watson
North Carolina State University

The IDGraphs intrusion detection system detects and analyzes a variety of attacks and anomalies, including port scanning, worm outbreaks, stealthy TCP SYN flooding, and some distributed attacks.

Traffic anomalies and attacks are commonplace in today's networks. Researchers estimate that malicious code caused more than $28 billion in economic losses in 2003, and will grow to more than $75 billion by 2007 (see http://www.mxlogic.com/pdf/About_MX_Logic.pdf). For these reasons, large network operators place great importance on rapid and accurate identification of traffic anomalies and attacks.

Most existing intrusion detection systems (IDSs) identify attacks using specific patterns in the attack traffic called *signatures*. But such IDSs cannot detect unknown network attacks, and attackers can easily foil detection by garbling their signatures. Other statistical IDSs use overall traffic to detect attacks, but suffer from inaccuracies and difficulties in finding attack flows, even when anomalies are correctly identified.[1] In addition, although a few flow-level detection schemes[2] monitor specific flows, the following questions remain open:

- Do intrusions such as TCP SYN flooding and port scans have characteristic time series patterns when observed from edge network routers? For instance, are there any common patterns for spread of a specific worm that might indicate its propagation strategy?
- How can we identify correlated attacks, especially when they are new? This is a difficult challenge for the intrusion detection (ID) community. To the best of our knowledge, almost all systems treat attacks independently, even after detecting the attacks.
- How can discovered intrusions and anomalies be analyzed interactively? Once an attack is detected, it must be examined closely to determine if it actually is an attack, in the broader context of other network traffic.
- What is the appropriate threshold for automatic statistical IDS? A good threshold will neither report too many attacks (false positives), nor too few (false negatives). This tradeoff is best examined in an interactive network context.

IDGraphs is an interactive visualization system that addresses these challenges, supporting intrusion detection over massive network traffic streams. It features a novel time versus failed connections mapping that aids in discovery of attack patterns. The number of failed connections (SYN-SYN/ACK) is a strong indicator of suspicious network flows.

IDGraphs offers several flow aggregation methods that help reveal different attack patterns. For example, to detect TCP SYN flooding, we aggregate flows with unique destination IP and port (destination IP, or DIP, and destination port, or Dport) pairs, collecting all data directed to a certain port on a certain machine.

The system also offers high visual scalability through the use of Histographs.[3] Users can view tens or hundreds of thousands of time series at once, with frequency of network events indicated by pixel brightness, and in-depth examination supported through a zooming interface.

IDGraphs' linked correlation matrix view reveals related attacks. Brushing in the matrix view reveals correlated flows in the IDGraphs view. To the best of our knowledge, we are the first to provide such views of correlated stream activity for intrusion detection. Another feature is the search and filter interface for ungraphed network data dimensions such as source IP (SIP) and Dport.

In this article, we demonstrate IDGraphs using a single day of NetFlow network traffic traces collected at edge routers at Northwestern University, which has several OC-3 links. These traces totaled 179 million records and 1.16 terabytes of traffic.

The "Previous Work" sidebar discusses other approaches to IDSs.

Threat model and data collection

Our ultimate goal is to detect as many attacks as possible. We begin by focusing on the two attacks of most concern in ID: denial-of-service (DoS) TCP flooding and port scans (mostly for worm propagation).

Threat model

Wang, Zhang, and Shin report that more than 90 per-

 Published by the IEEE Computer Society

Previous Work

With the rapid growth of network bandwidth and fast emergence of new attacks and worms, there is a growing body of research in both automated intrusion detection systems (IDSs) and applications of visualization for intrusion detection.

Intrusion detection systems

An IDS is a type of security management system for computers and networks. It automatically gathers and analyzes information from various areas within a computer or a network to identify possible security breaches, which include both intrusions and misuse. Many IDSs such as Bro[1] and Snort[2] check packet payloads for virus or worm signatures. However, such schemes don't scale to high-speed network links and can't detect previously unknown types of attacks. For these reasons, many researchers have proposed techniques based on the statistical characteristics of the intrusions.

We classify these techniques into two rough categories: detection based on overall traffic[3] such as change point monitor, which tends to be inaccurate and can't find real attack flows; and flow-level detection such as threshold random walk, which is vulnerable to denial-of-service attacks with randomly fabricated (spoofed) IP addresses. Flow-level detection is especially vulnerable on high-speed networks since the sequential hypothesis testing scheme it uses needs to maintain a per-SIP table for detection. Gao et al. recently addressed this problem using a reversible sketch technique.[4]

Most ID technologies perform detection on individual traffic flows, rather than looking for the correlations between multiple flows. These methods can only provide a small snapshot of globally distributed attacks. More recently developed correlation information analyses address this problem, reducing the high volume of alerts and false positives.[5]

Visualization for Internet security

In applying visualization to Internet security, researchers exploit the innate human ability to process visual information quickly, enabling the complex tasks of network security monitoring and intrusion detection to be performed accurately and efficiently. Many systems have addressed this problem.[6-10] All of them provide interactive visual support for anomaly detection.

PortVis produces visualizations of network traffic using 2D plots on Dport versus Dport axes (the axes display complementary Dport subfields).[8] It summarizes network activity at each location in the plot using color. Users can drill down to display traffic information at finer temporal and port resolutions, ultimately reaching explicit Dport numbers and traffic levels. Temporal NetFlow profiles are also available.

VisFlowConnect uses a simple application of parallel coordinates to display incoming and outgoing network flows.[9] SIPs are displayed on a left vertical axis, local subnet IPs on a central vertical axis, and DIPs on a right vertical axis. Individual flows appear as two-segment polylines spanning the three parallel axes. Flow volume is mapped to line segment thickness.

The Spinning Cube plots traffic in three dimensions on SIP versus DIP versus Dport axes.[7] The amount of network activity is visualized interactively using color, displaying certain attacks (especially port scans) clearly.

NVisionIP[6] visualizes network flow in a subnet versus host (DIP high bits versus DIP low bits) matrix. Each matrix cell represents the traffic destined for a certain host. Users can reduce or increase viewed detail. As users navigate through the visualization to make discoveries, NVisionIP records the implicitly formed queries into a tree structure using a symbolic language. These records might then be handed off to an IDS for automated detection.

IDS RainStorm visualizes alarms generated by an IDS using a time versus DIP axes, with color showing the severity of alarms.[10] Zooming provides more detail within certain DIP and/or time ranges.

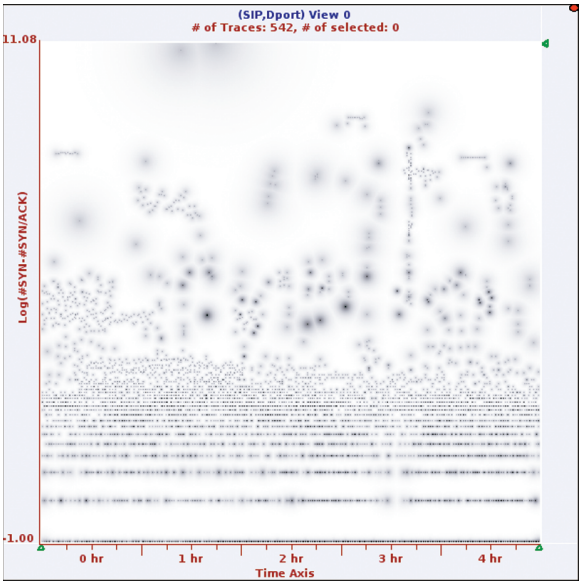
While all of these systems have strengths, none of them addresses all four of the questions we posed in our introduction. We make detailed comparisons of IDGraphs to these systems in the "Comparison to ID visualization systems" section in the main article text.

References

1. V. Paxson, "Bro: A System for Detecting Network Intruders in Real Time," *Computer Networks*, vol. 31, no. 23-24, 1999, pp. 2435-2463.
2. M. Roesch, *Snort: The Lightweight Network Intrusion Detection System 2001*; http://www.snort.org/docs/lisapaper.txt.
3. D. Moore, G. Voelker, and S. Savage, "Inferring Internet Denial of Service Activity," *Proc. USENIX Security Symp.*, USENIX, 2001, pp. 9-22.
4. Y. Gao, Z. Li, and Y. Chen, "A DoS Resilient Flow-Level Intrusion Detection Approach for High-Speed Networks," to be published in *Proc. 26th Int'l Conf. Distributed Computing*, 2006.
5. C. Abad et al., "Log Correlation for Intrusion Detection: A Proof of Concept," *Proc. Ann. Computer Security Applications Conf.* (ACSAC), IEEE CS Press, 2003, pp. 255-264.
6. K. Lakkaraju, W. Yurcik, and A. J. Lee, "NVisionIP: Netflow Visualizations of System State for Security Situational Awareness," *Proc. ACM Workshop Visualization and Data Mining for Computer Security* (VizSEC/DMSEC), ACM Press, 2004, pp. 65-72.
7. S. Lau, "The Spinning Cube of Potential Doom," *Comm. ACM*, vol. 47, no. 6, 2004, pp. 25-26.
8. J. McPherson et al., "Portvis: A Tool for Port-Based Detection of Security Events," *Proc. ACM Workshop Visualization and Data Mining for Computer Security* (VizSEC/DMSEC), ACM Press, 2004, pp. 73-81.
9. X. Yin et al., "VisFlowConnect: Netflow Visualizations of Link Relationships for Security Situational Awareness," *Proc. ACM Workshop Visualization and Data Mining for Computer Security* (VizSEC/DMSEC), ACM Press, 2004, pp. 26-34.
10. K. Abdullah et al., "IDS Rainstorm: Visualizing IDS Alarms," *Proc. IEEE VizSEC* Visualization for Computer Security, IEEE CS Press, 2005, pp. 1-10.

Table 1. The selectivity of different types of aggregation keys. The bottom three single-field keys are less selective.

Key Types	SYN Flooding	IP Scan	Port Scan	Hybrid Scan
(SIP, Dport)	Nonspoofed only	Yes	No	Yes
(DIP, Dport)	Yes	No	No	No
(SIP, DIP)	Nonspoofed only	No	Yes	Yes
(SIP)	Nonspoofed only	Yes	Yes	Yes
(DIP)	Yes	No	Yes	Yes
(Dport)	Yes	Yes	No	Yes



1 NetFlow streams aggregated by (SIP, Dport) for detection of IP scans.

cent of DoS attacks are TCP SYN flooding attacks.[4] Although DoS might also include corruption attacks (see http://www.cert.org/tech_tips/denial_of_service.html), we exclude them here because they are often application or protocol specific. Scans are probably the most common and versatile type of intrusion.

Three well-known types of scans exist: IP or horizontal scans, port or vertical scans, and their hybrid block scans.[5] (Because we do not use the traditional DIP versus Dport mapping, we prefer the terminology IP, port, and hybrid scans here.) Unlike DoS attacks, scans must use a real SIP, because the results of the scan (open ports or responding IPs) must be returned to the attacker.[5] IP scans are most common—they scan certain ports across an interesting range of IPs. These certain ports reflect the vulnerability the attackers try to exploit. A port scan queries ports on a single host, usually because the attacker is interested in this particular host, and wishes to characterize its active services to find which exploits to attempt.[5]

Data collection and aggregation

IDGraphs is based on preprocessed NetFlow data, but it is simple to extend to other data sources. NetFlow data was originally derived from Cisco routers caching recent flows for lookup efficiency, and it has now become the de facto standard for router traffic monitoring accepted by all other major router vendors. NetFlow is a unidirectional stream of packets between a given source and destination both of which are defined by a network-layer IP address and transport-layer source and destination port numbers. Here we only consider the attacks in TCP protocol—in other words, the TCP SYN flooding attacks and TCP scans. We analyze the attributes in TCP/IP headers and select a small set of metrics for flow-level traffic monitoring. The possible fields we can use are DIP, SIP, Dport, and source port (Sport). Because source port can be chosen arbitrarily, it's not useful for attack detection. We could aggregate the traffic by all combinations of the remaining three fields, but grouping by the key (SIP DIP, Dport) would break up all attacks except nonspoofed SYN flooding, so we don't use it in detection. Table 1 shows the other key combinations and their sensitivity to different types of attacks.

We deem keys sensitive only if a unique key will capture most of an attack within its corresponding aggregated data group. In general, single-field keys are less selective for specific types of attacks than two-field keys since they are each sensitive to several attack types.

IDGraphs design

We built IDGraphs on top of the Histographs visualization system,[3] with enhancements designed specifically for visualizing NetFlow data sets. The data input can be aggregated using any of the six keys listed in Table 1, though we find the two-field keys more useful. In preprocessing we sort NetFlow records by key and then time to form a time series for each key. We filter out streams with less than five unsuccessful connections over the whole time range.

IDGraphs helps Internet security experts inspect their NetFlow data visually and perform deep analysis. Users can quickly identify possible anomalies or attacks using overviews then follow up with in-depth analyses by querying those possible anomalies. Figure 1 shows an overview of traffic aggregated by (SIP, Dport) for detection of IP scans. The horizontal axis indicates time, while the vertical axis indicates number of failed connections.

Dark points in Figure 1 indicate high data density; we used splatting (blurring) to increase the visibility of isolated points with abnormally high failure counts. Figure 2 shows in-depth querying revealing that the dark dots in hours 1 and 2 are hybrid scans, which are probing multiple Dports, and with failure counts of 111, aggregated across multiple DIPs.

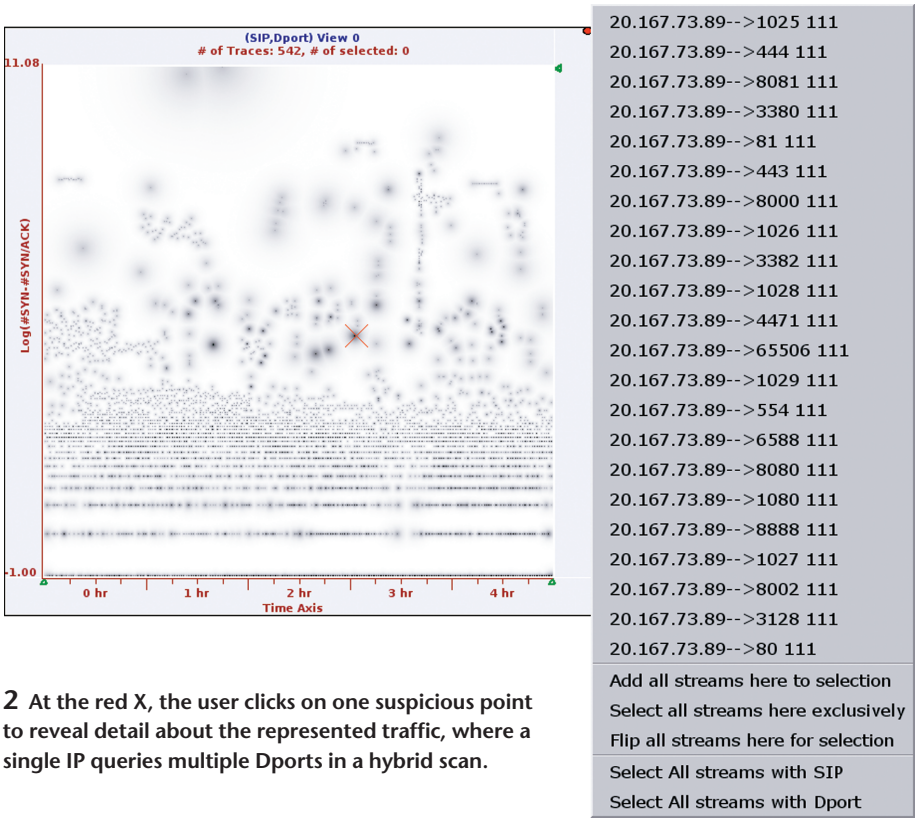
Visual mapping

Unlike most previous systems, IDGraphs displays time series data—a temporally ordered sequence of failed connection counts (SYN-SYN/ACK) for each aggregating key. The resulting time versus failed connection mapping is quite powerful, aligning the vertical display dimension with a strong indicator of suspicious traffic. Thus, the higher a point is in the IDGraph, the more suspicious the traffic it represents. In Figure 1, the user has

transformed the failure count using a log function, compressing the data vertically and making more efficient use of display space. Points at the top of the IDGraph are likely candidates for further user scrutiny, while the darker, horizontal structures at the bottom of the graph depict the large majority of normal traffic. The log transform organizes this trustworthy normal traffic into regular linear structures, making it easy to identify and ignore.

The number of NetFlow streams we can view at once is only limited by available display space and machine memory. We face an occlusion problem in display space: multiple data points can be mapped to the same display pixel. Our base Histographs system[3] plots dense and high-dimensional data by stacking or compositing graphs, and addresses this problem with a number of techniques. First, similar to the Information Mural system,[6] our system maps the number of data points at a pixel (frequency or data density) to pixel luminance, darkening those regions of the plot where data is dense. This highlights the main data trends but unfortunately, it also makes it difficult to perceive outliers. Histographs addresses this problem in two ways. First, it introduces a new, contrast-weighted mapping between data and luminance that highlights changes in data frequency. Second, when data points are isolated, it adds lower spatial frequencies to them to increase their visibility (splatting) without adjusting the data-luminance mapping.

These measures are particularly important in IDGraphs, where outliers are precisely what users are seeking.



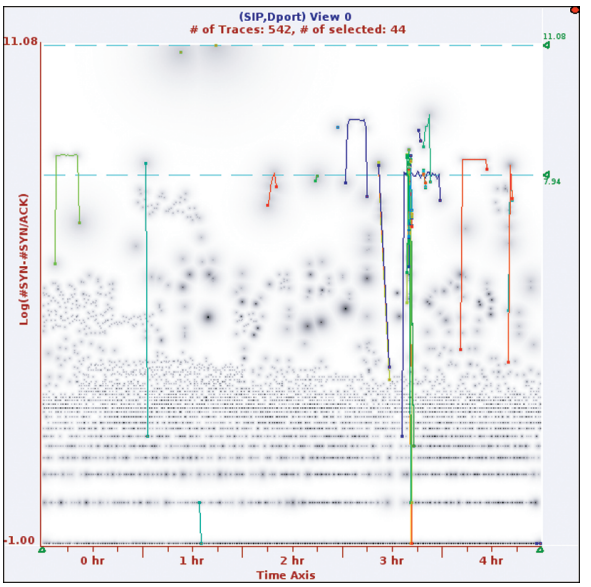
2 At the red X, the user clicks on one suspicious point to reveal detail about the represented traffic, where a single IP queries multiple Dports in a hybrid scan.

Interactive query

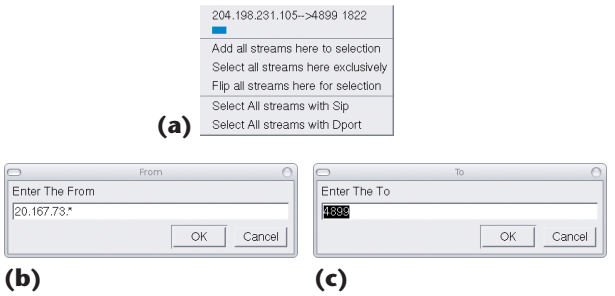
Interaction is the key to performing deep analysis with IDGraphs. Our design is guided by Shneiderman's visual information-seeking mantra,[7] aiming to provide detailed information whenever the user asks for it. The dynamic query techniques pioneered by Shneiderman also heavily influenced our design. The ability to click and query is central to interactive analysis with IDGraphs. Clicking for selection is tolerant of inaccuracy, allowing a 1-pixel mismatch between the location of nearby data and cursor location.

This is especially effective when the user wants to query an isolated data pixel. In Figure 2, the user clicks on a pixel to reveal a pop-up menu showing textual information about the data from different streams aggregated by this pixel. This reveals the specific keys and streams mapped to this pixel, and the associated failure counts. Users can use this menu to select some or all of the streams highlighting those streams in the IDGraph. Users can also use axis sliders to select streams meeting certain temporal or failure thresholds.

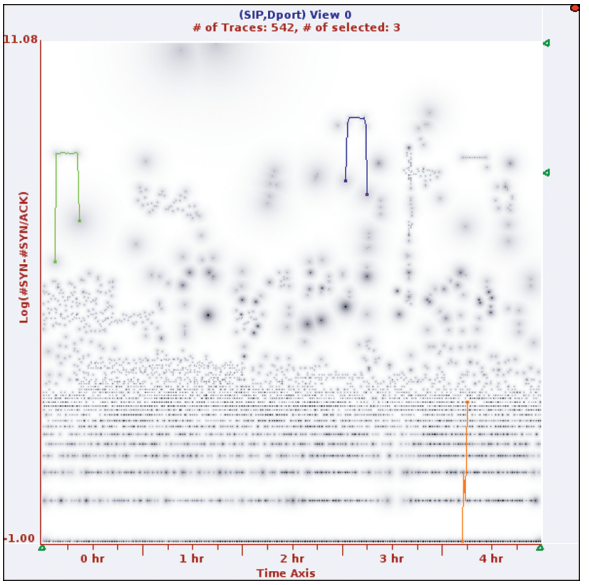
In Figure 3, the user selects all streams failing to connect more than 1,000 times at least once in the displayed period. We highlight selected streams in the IDGraph by linking the data points of each selected time series with lines. Different colors are applied to each stream. Stream data might not be contiguous; in such situations the streams appear as several disconnected polylines, with filled circles emphasizing the start and the end point of each trace. Currently, selected streams are indicated in the query interface by color bars that have the same color as the lines highlighting the streams in the IDGraph itself.



3 The user selects streams failing to connect more than 1,000 times.



4 A query interface allows selection of streams with certain SIPs, DIPs, and/or Dports.



5 The user selects all the streams with destination port 3306, which services MySQL.

To provide real-time intrusion detection, IDS systems often use default-detection thresholds to identify suspicious network activity. These thresholds are important in both simulation and actual detection. Determining such thresholds is difficult. The failure sliders of Figure 3 let users study possible detection thresholds interactively using brushing, with immediate visual feedback.

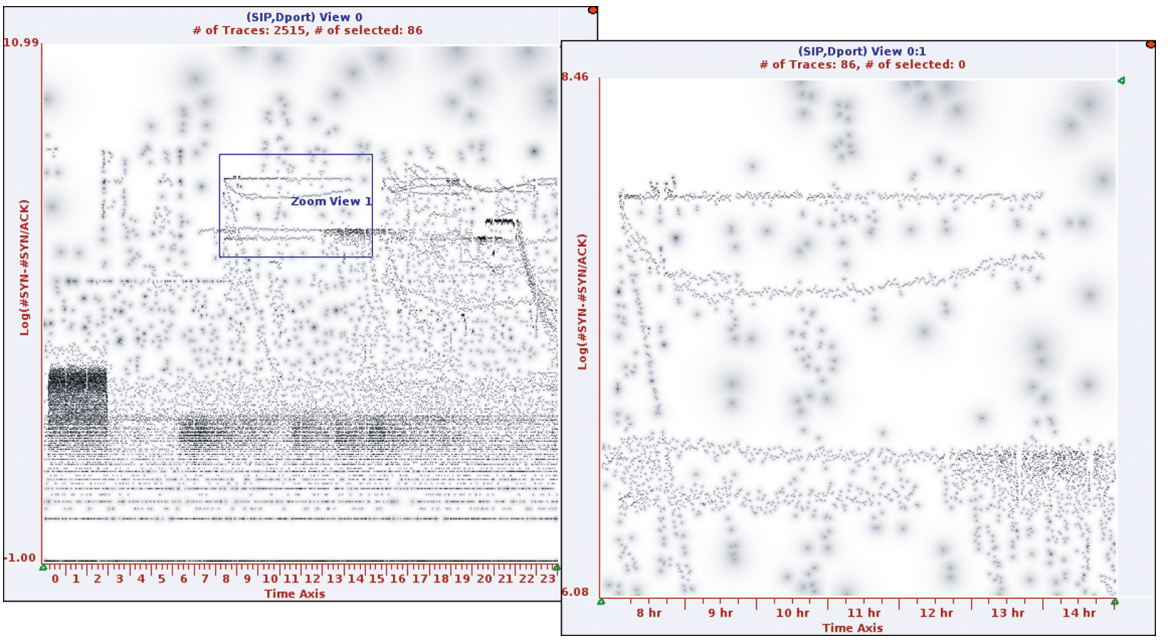
Having found suspicious network activity, users will often try to generalize the discovery by searching for other streams with similar features. To address this problem, we provide a more general query interface that lets users select streams with the same or similar (using wildcards) source and destination IPs. Figure 4 shows the interface as accessed after a point-and-click interaction. Users can also access this query interface directly, without first clicking. In Figure 5, the user employs the interface to select all streams with Dport 3306, which services MySQL.

Users can also annotate particular discoveries in an IDGraph so that they and colleagues can quickly find them later for further discussion and analysis. By default only, a red dot is visible; clicking on the dot reveals annotation text.

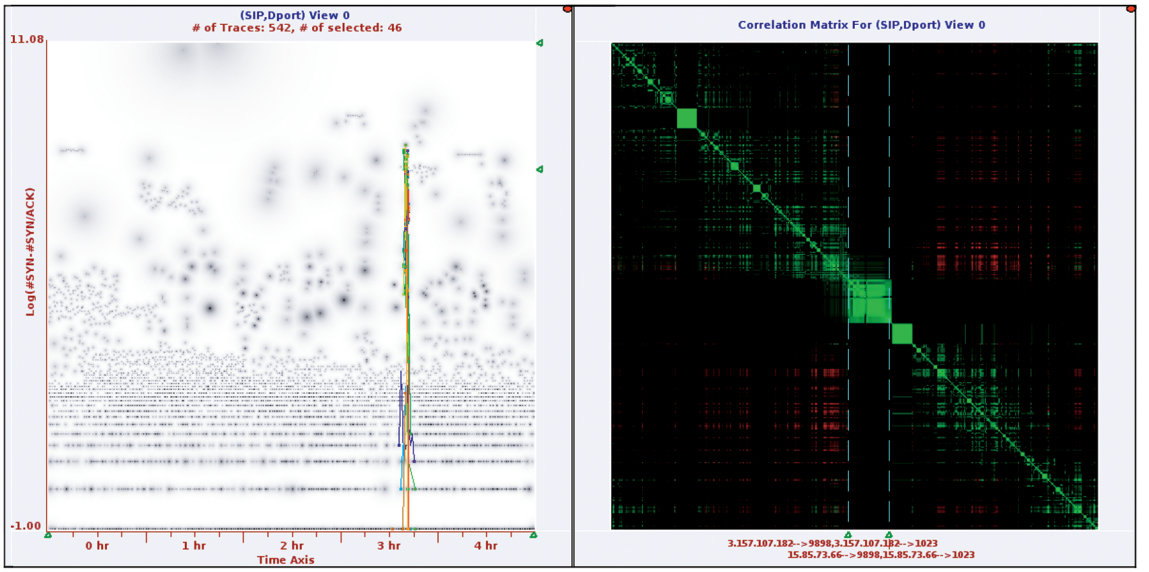
Data space zoom

Having seen an overview, users can zoom in on interesting data features, obtaining more detailed information about the displayed traffic. In Figure 6, the user draws a rectangle over a region of interest simultaneously selecting a time and failure count range. For more precision in selection, sliders might also be used. When the selection is complete, a new view appears, displaying the selected range of traffic, and any streams that intersect this range. Zooms might also be performed on zoomed views. To maintain a visual correspondence between a zoomed view and its parent context, we use the same frequency-to-luminance mapping from the parent view.

To generate zoomed views, we organize NetFlow data into a detail pyramid, with higher temporal resolutions used at the finer, more detailed levels. Each zoomed



6 Zoomed view on the right reveals detail highlighted on the left.



7 Brushing with a linked correlation matrix view to reveal coincident IP scans from several different SIPs.

view uses the finest temporal resolution supported by the current display resolution.

Correlation analysis

To help users form and test hypotheses about relationships between two or more NetFlow streams, or simply to identify streams with similar temporal NetFlow signatures, IDGraphs provides a linked correlation matrix view. Figure 7 shows this view, which takes the form of a matrix with each aggregated group of NetFlow streams represented by one row and one column. When the number of keys/groups is greater than the number of rows/columns, each row and column represents several groups. Each cell in the matrix represents the correlation between two sets of groups, with red indicating a negative correlation and green positive, while luminance increases with the correlation's absolute magnitude. In Figure 7, the user brushes one of the largest correlated blocks, selecting the streams forming the vertical, linear structure in the linked IDGraph (see Figure 1). These coincident attacks are IP scans issued from a number of different SIPs, primarily targeting three ports.

In Figure 8 (next page), the user is visualizing five hours of NetFlow traffic aggregated using (DIP, Dport) keys to detect SYN flooding attacks. Using two sliders in the IDGraphs view, the user chooses a two-hour time range over which to construct a correlation matrix and then selects a small cluster in the matrix itself, revealing four probable, coincident attacks with similar patterns, each targeting a different machine. These attacks are well below typical automatic IDS thresholds, and would be difficult to detect using traditional ID methods.

Selection in the correlation view would be difficult and have little purpose, were correlated streams distributed widely across the matrix. We avoid this problem by reordering NetFlow streams in the matrix into correlated clusters. To perform this reordering, we apply the correlation matrix ordering technique that Friendly describes.[8] We treat each row (column) in the matrix as a point in a high-dimensional space, and apply principal component analysis. Each row (column) is then projected into the 2D space that the first two eigenvectors of the correlation matrix describe. We then order radially these projected 2D points and apply the same ordering to the rows (columns) of the correlation matrix.

Case studies

Here we describe several examples of the use of IDGraphs for network anomaly detection.

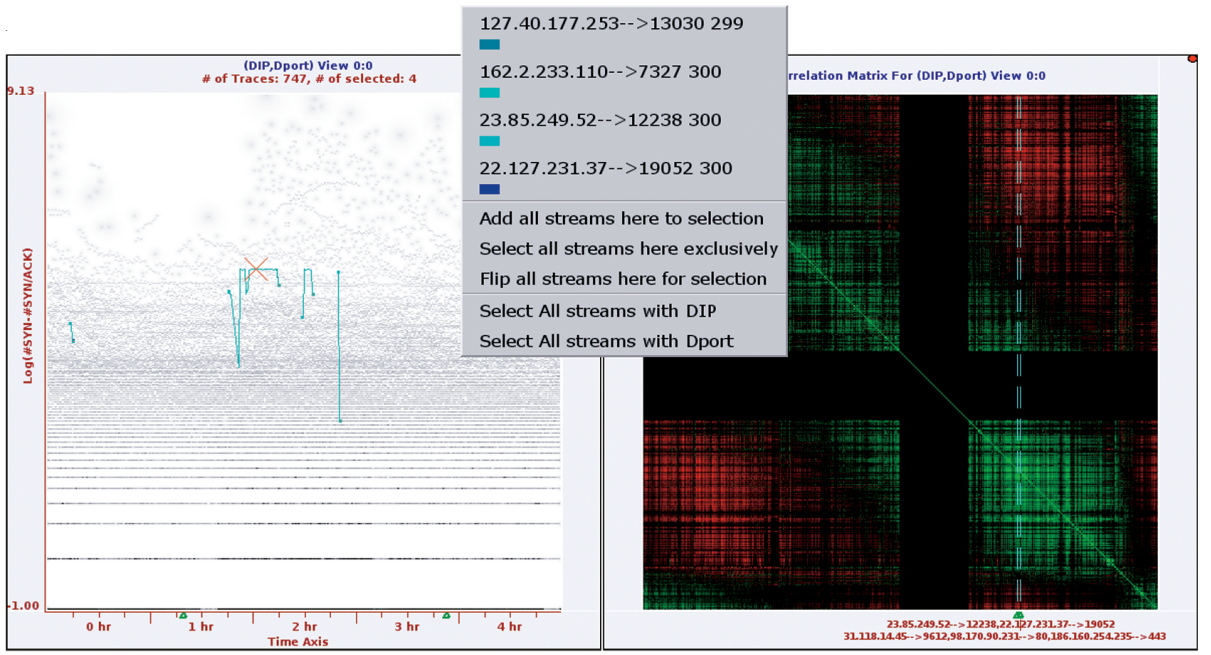
IP scan from a coordinated worm attack

Figure 1 visualizes five hours of NetFlow data aggregated with (SIP, Dport) keys. In the middle of hour 3, we see a suspicious vertical linear structure. In Figure 3, we select the streams that reach the same failure count, revealing many streams with failures restricted almost exclusively to the time range spanned by this linear structure. Clicking on these streams reveals that they are from different SIPs, but communicate with three common destination ports: 5554, 9898, and 1023. The Dabber backdoor and Sasse worms target these ports . We discovered these coordinated attacks without prior knowledge of this port information.

Having identified these suspicious ports, we can select the streams connecting to those ports via the query interface shown in Figure 4 quickly identifying all the possible attacks by this worm within our data set, even if they are smaller and stealthier. Because they are highly similar, these streams are also salient in Figure 7's correlation matrix view, appearing as the large green block in the middle of the matrix.

Hybrid scan and temporal similarities in IP scans

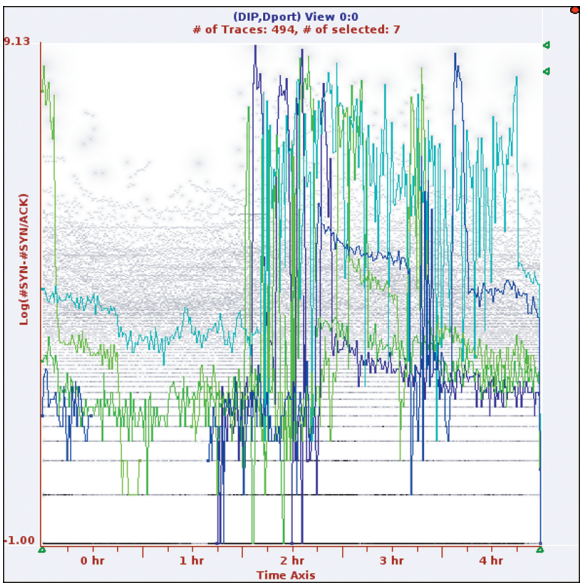
Those streams with a high number of unsuccessful connections in the data aggregated by (SIP, Dport) and shown in Figure 2 are possible IP scans. We can automatically detect such streams using good thresholding. However, IDGraphs allows for an immediate deeper



8 Correlation brushing for SYN flooding attacks in data aggregated using (DIP, Dport) keys reveals four coincident probable SYN flooding attacks.

analysis. The suspect streams appear as several dark, splatted dots. By clicking on them, the user can reveal detailed textual information. In this case, we learn that all these streams are from the same SIP and target different Dports, that is, a port scan. Since it is unlikely that the SYN-SYN/ACK failure count would be high for each of these streams if they each only addressed one DIP, the attack is likely also an IP scan, and therefore also probably a hybrid scan.

Figure 3 highlights several suspicious streams. In particular, notice the two similarly shaped streams at the beginning of hour 0 (light green) and the beginning of hour 3 (blue). Clicking on them, we find that they both communicate with Dport 3306, which MySQL uses.



9 The seven most suspicious SYN flooding attacks selected in a visualization aggregated by (DIP, Dport).

These two possible attacks share the same temporal pattern; note especially the almost constant connection failure rate to the MySQL database for a time period of 15 to 20 minutes. We suspect this pattern might indicate a consistent hacking technique—perhaps password guessing. By querying and selecting all the streams with this Dport as shown in Figure 5, users can further examine all suspicious communication with MySQL in the data set.

SYN flooding pattern discovery

Theoretically speaking, any streams with high failure values in the (DIP, Dport) data set are potential TCP SYN flooding attacks. But IDGraphs lets users pursue this initial hypothesis more deeply.

Figure 9 reveals the temporal patterns of the most suspicious NetFlow streams and shows that they had SYN-SYN/ACK values that peaked during hours 2 and 3. In Figure 8, we brush on a linked correlation matrix view to reveal four streams with similar temporal patterns. Even though the DIPs and Dports for these streams are totally different, it's highly probable that these flooding attacks emanate from the same source. In Figure 10, we test this hypothesis by visualizing the same traffic keyed and aggregated by (SIP, DIP).

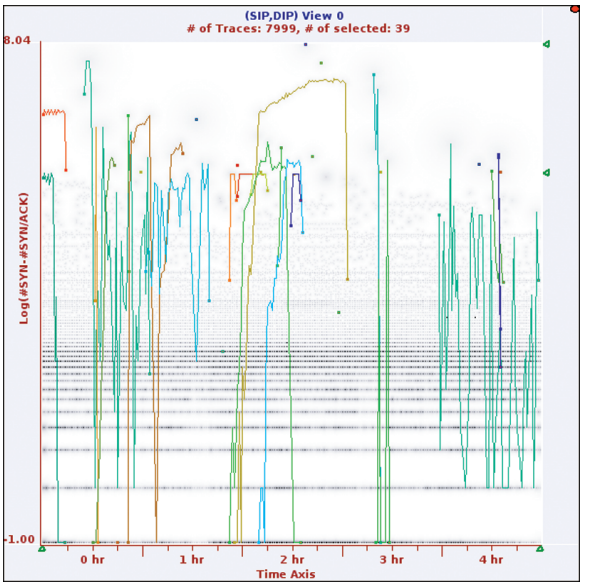
Querying for and highlighting streams with these four DIPs, we find that at any given time the attacks indeed emanated from the same SIP. While SIPs did change over time, they were always from the same subnet. It seems the attacker was flooding destination hosts on a list and trying to hide his attack by switching the SIP from time to time.

Worm propagation pattern discovery and strategy inference

Using IDGraphs time-series visualization, patterns in anomalous activity patterns are simple to spot.

This offers clues about the propagation strategy of the associated attacks. For instance, we found a regular series of periodic IP scans to TCP port 25 (servicing SMTP) as illustrated in Figure 11. It appears to result from the RTM Sendmail Worm. The infected host sends out a burst of scan packets periodically, with each period having nearly the same minimum (0) and maximum (800) SYN-SYN/ACK values, likely to avoid overloading the attacking machine and its network bandwidth.

When we highlight all the traffic to port 25, we see the results in Figure 12a (next page). Several of the most suspicious streams seem to share a similar temporal pattern, though this pattern is different from the pattern in Figure 11. Could this new pattern indicate a different, coordinated attack? In Figure 12b, we click and query to find the source IP of the most suspicious scans and follow up with a search by source IP subnet in Figure 12c, learning that indeed, they are from the same subnet.



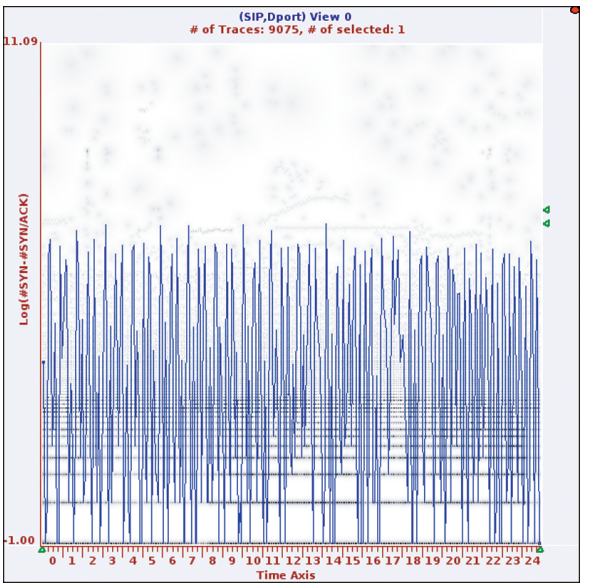
10 The most suspicious attacks selected in a visualization aggregated by (SIP, DIP).

Comparison and evaluation

We evaluated our IDGraphs system with comparisons to existing automated and visualization-based ID systems and through a discussion with a practicing computer systems administrator.

Comparison with an automated IDS

The High-Speed Router-Based Anomaly and Intrusion Detection (HRAID) system is a traditional IDS developed at Northwestern that uses automation to detect intrusions in real time.[9] Its primary detection mechanism is the identification of network flows with unusually high failed connection counts. HRAID also features hash-table-(sketch)-based data aggregation enabling low memory consumption and utility with high-bandwidth flows. We were inspired by HRAID in designing IDGraphs; we used an intermediate HRAID NetFlow data format as input. IDGraphs also uses failure count as a primary detection measure and, like HRAID, aggregates data with (SIP, Dport), (DIP, DPort), and (SIP, DIP) keys. As a result, the same flows HRAID identifies as attacks are displayed at the top of the SYN-SYN/ACK axis in IDGraphs. Nevertheless, there are several important differences between HRAID and IDGraphs. The first are the design goals. HRAID targets automated, real-time detection of network anomalies and attacks; rapid and accurate reaction to those attacks is the key to its success. We designed IDGraphs to complement HRAID by

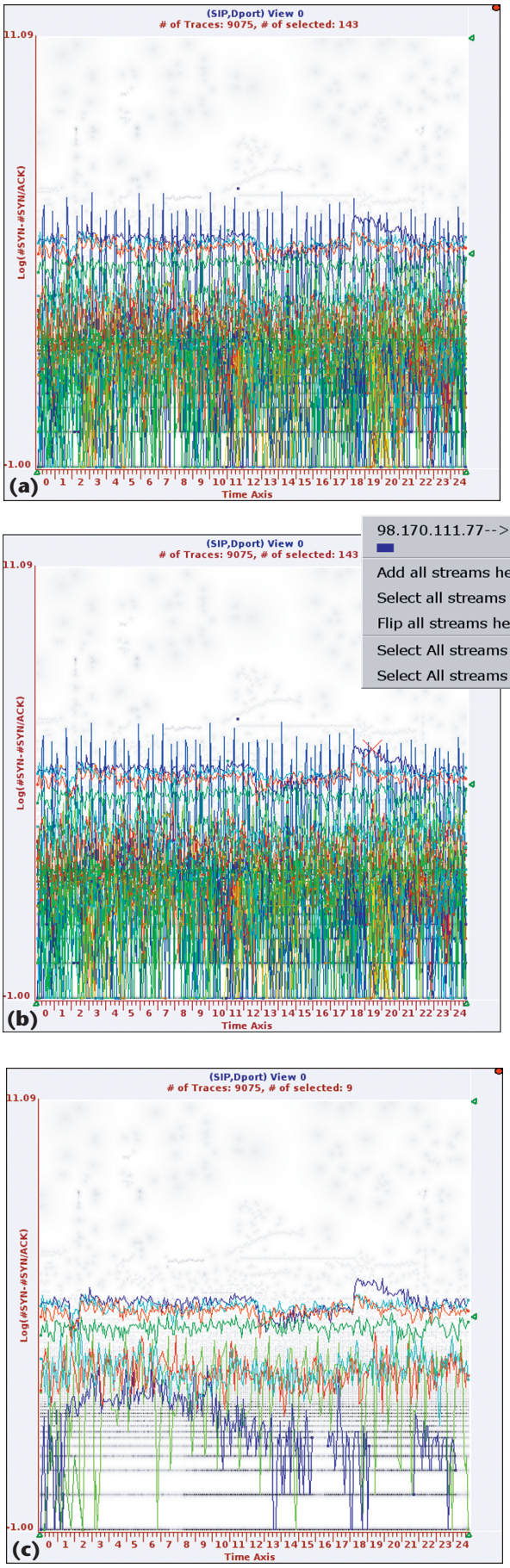


11 The temporal pattern of an IP scan to port 25 revealed over a 25-hour period in traffic aggregated by (SIP, Dport).

- providing visual analysis and validation of attacks and anomalies detected by HRAID;
- capturing and visualizing the temporal patterns in anomalous network traffic; and
- exposing undetected attacks that might be detected in HRAID using new heuristics and providing the information needed to construct those heuristics.

Visual analysis and validation. This should be particularly important as system administrators tune HRAID or other automated ID systems to their local networks, setting the statistical thresholds above which flows are treated as attacks and finding the optimal tradeoff between false negatives and positives. Figure 13 visualizes the 10 most suspicious IP scans detected in HRAID. We selected these scans in IDGraphs with a threshold slider, in one day's Northwestern University NetFlow data set aggregated by (SIP, Dport). These scans are sudden bursts of (failed) connections, represented as isolated points in the IDGraph. We might use visualizations such as these to tune automated detection thresholds.

Visualizing temporal patterns. These temporal patterns are an underutilized component of attack signatures. Statistical maxima are often not enough to identify attacks. Attacks might spread their traffic across time and/or IPs, effectively flying beneath the radar of threshold-based detection schemes. For such attacks,



12 (a) All the traffic to port 25 in the same period. Several of the most suspicious streams have similar temporal patterns. (b) Clicking and querying on one of the most suspicious streams reveals it comes from SIP 98.170.111.77. (c) A query with wildcards showing all the streams issued from subnet 98.170. All the similarly patterned streams come from this subnet and are probably correlated attacks.

alternative detection signatures must be found. Figure 11 shows an attack that would be below most thresholds, and yet exhibits temporal patterns that might be useful in automated detection.

Exposing undetected attacks. In Figure 13, the dark block in hours 0 through 2 is well below automated thresholds. In Figure 14, we select and query this block to reveal that it's formed by streams from multiple SIPs to ports 6129 and 1433 in common DIP subnets. It's highly probable that these are coordinated backdoor attacks launched from infected or compromised hosts trying to exploit the vulnerability of other hosts. Given the pattern discovered in this visualization, automatic IDS might be improved to detect such attacks by summing the flows keyed by destination subnets and certain Dports over certain limited temporal and failure count ranges.

Comparison to ID visualization systems

IDGraphs distinguishes itself from existing ID visualization systems in three primary respects. First, its use of a failure count versus time–spatial mapping makes traffic anomalies and their temporal signatures visually apparent. Second, its use of Histographs to composite NetFlow streams drastically improves this approach's scalability. Finally, its use of a wide range of NetFlow data fields including SIP, DIP, Dport, time, and SYN-SYN/ACK gives a broad view of network activity.

Comparison to PortVis. The Dport versus Dport mapping used by PortVis[10] does not highlight suspicious activity quite as clearly as IDGraphs time versus failure-count mapping. Although PortVis offers a temporal view for examining temporal patterns, this view is not the application's focus, and it's difficult to compare different flows to find correlated attacks. IDGraphs' support for viewing NetFlow traffic aggregated by port is not as intuitive as such support in PortVis.

Aggregating data using (SIP, Dport) or (DIP, Dport) keys or querying by Dport allows IDGraphs users to visualize traffic targeting certain ports. Nevertheless, while such views highlight suspicious streams with their failure-count mapping, identifying the specific Dport being targeted requires interaction not always necessary in PortVis.

Comparison to VisFlowConnect. VisFlowConnect's locally centered parallel coordinates mapping provides an excellent overview of the source and destination of local traffic.[11] If the user is familiar with the typical patterns of this traffic, drastic changes in these patterns will be reason for suspicion.

Still, subtler and correlated attacks will be difficult to spot especially as the number of flows increases and the line segments in the visualization begin

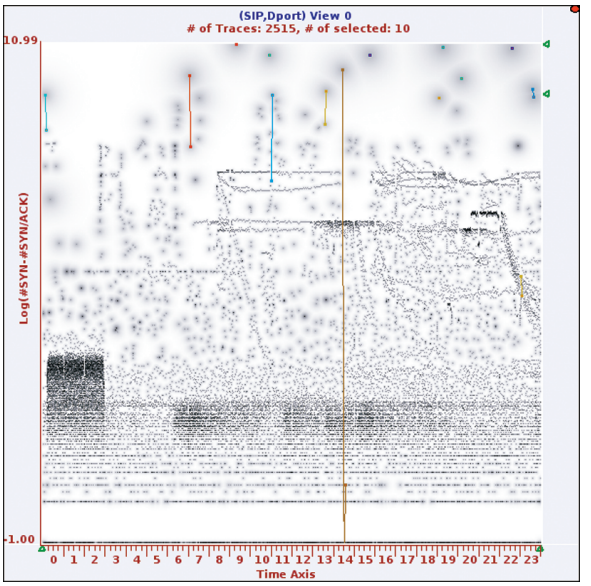
to occlude one another. IDGraphs addresses these problems with its failure-count mapping, which highlights suspicious traffic even when it's low in volume; and with its stream compositing methods, which greatly ameliorate the occlusion problem and increase scalability. With its time mapping, IDGraphs makes temporally correlated activity quite visible, facilitating the identification of correlated attacks. However, IDGraphs time versus failure-count mapping is not as successful in providing an overview of the sources and directions of local network flows. VisFlowConnect users will have a much better notion of how traffic volume relates to the local network.

Comparison to NVisionIP. NVisionIP provides a good overview of the direction (not the source) of network traffic.[12] By zooming in, users can focus on traffic directed toward individual subnets, ultimately visualizing traffic directed toward individual machines. When focused on the user's own subnet, NVisionIP supports monitoring of the traffic arriving at local hosts, making anomalies significant at this subnet level quite visible, especially when they are directed toward relatively quiet hosts. Once more however, stealthier, coordinated attacks will be more difficult to spot than with IDGraphs. For example, a low-volume attack on an active host would not change NVisionIPs visualization much, while a similar attack in IDGraphs would likely be visualized in isolation toward the top of the failure count axis.

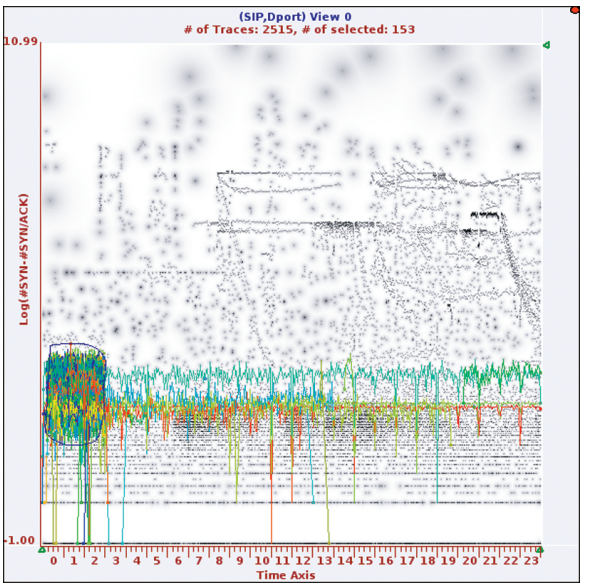
IDGraphs doesn't provide as effective an overview of network activity by IP address as NVisionIP. However, users can easily highlight all the streams to or from a certain IP, and then segregate the especially suspicious ones from more benign streams using the failure count axis. IDGraphs shares many of the same design considerations as NVisionIP: visualization should be used together and finally integrated with other analysis and detection tools. Transforming visual patterns to symbolic rules as in Lakkaraju, Yureik, and Lee[12] is a good step in this direction. In IDGraphs, we attempt this by visualizing measures and parameters already familiar to ID practitioners, such as detection thresholds and temporal correlations.

Comparison to IDS RainStorm. IDS RainStorm visualizes the results of automated ID, making it ideal for monitoring and responding to known attacks and exploits.[13] IDGraphs visualizes the NetFlow itself, making it sensitive to both known and unknown attacks. IDS RainStorm does not immediately visualize the attack type; the user must interact further to reveal this information. IDGraphs always displays temporal patterns specific to each attack, though these patterns will often be unfamiliar.

To make maximal use of display space, IDS RainStorm splits the vertical DIP dimension into several segments and spreads these horizontally across the display, resulting in several time versus DIP plots; scaling this dense display to more than a few thousand hosts would be difficult. With stream compositing, IDGraphs scales more gracefully as the number of hosts increases. IDS RainStorm's time versus DIP mapping more immediately depicts the targets and history of any attacks (although its segmentation of the DIP address space requires learning). On the other hand, IDGraphs time versus failure-count mapping more effectively conveys the NetFlow activity of attacks.



13 One day of NetFlow traffic aggregated by (SIP, Dport) and then selected using a threshold slider. The 10 highlighted scans are the same detected by the automated IDS HRAID.



14 Clicking reveals that many of these coordinated attacks target port 6129.

Practitioner feedback

We demonstrated our system to a practicing network administrator and received several comments and suggestions. To the administrator, the utility of IDGraphs in explaining the characteristics of identified attacks to managers and colleagues was obvious. He also confirmed that IDGraphs visualizations should be quite useful in validating and tuning automated ID tools.

In the future, the administrator would like to see IDGraphs working with live NetFlow data, perhaps by

introducing a horizontal roll to the display. IDGraphs and automated ID tools such as HRAID might be more closely integrated, so that the likelihood that a flow is an intrusion trace might be displayed (instead of a simple binary threshold). Finally, the correlation function might be expanded so that once a suspicious stream is identified, others like it might quickly be found.

Conclusions and future work

IDGraphs is an interactive system for visualizing NetFlows, capable of detecting network anomalies and attacks including port scans, worm attacks, and SYN flooding. Perhaps more importantly it can lead to useful insights concerning the propagation and intrusion patterns used in network attacks, even if they are distributed or spoofed. While IDGraphs uses a time versus failure count plot, most other ID visualization systems use plots based on IP address and/or port. Such address-based mappings are useful, and IDGraphs should complement them well.

IDGraphs is certainly not without its limitations. Although we have reduced visual clutter by our frequency-to-luminance mapping, clutter and occlusion are still quite evident when many streams are highlighted (as in Figure 12b). One solution we have already implemented uses a constant hue for highlighted streams and our regular frequency-to-luminance mapping to address occlusion. This does not, however, allow users to distinguish one highlighted stream from another.

We have already mentioned that although queries by individual SIP, DIP, and Dport are available, our current spatial mapping does not provide a good overview of the data distribution across these fields. An improved overview might be provided by offering SIP, DIP, and Dport sliders that let users page through the streams partially keyed by these fields. Also, because Histographs can function with any spatial mapping, it might be interesting to experiment with linked SIP, DIP, Dport, or traffic volume versus time views.

While our correlation view is valuable, it's only useful when coordinated attacks have well aligned temporal patterns. We plan to experiment with correlation techniques that do not require such alignment, and perhaps also with frequency-space transformations of temporal patterns. ■

Acknowledgments

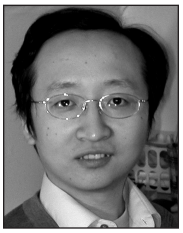
This research was supported by National Science Foundation grant 0093172. We thank Peter Dinda for his suggestions and John Kristoff for his practitioner's viewpoint.

References

1. D. Moore, G. Voelker, and S. Savage, "Inferring Internet Denial of Service Activity," *Proc. USENIX Security Symp.*, USENIX, 2001, PP. 9-22.
2. V. Paxson, "Bro: A System for Detecting Network Intruders in Real Time," *Computer Networks*, vol. 31, no. 23-24, 1999, pp. 2435-2463.
3. P. Ren and B. Watson, *Histographs: Interactive Visualization of Complex Data with Graphs*, tech. report NWU-CS-05-12, Dept. Computer Sciences, Northwestern Univ., 2005; http://www.cs.northwestern.edu/publications/techreports/2005_TR/NWU-CS-05-12.pdf.
4. H.N. Wang, D.L. Zhang, and K.G. Shin, "Change Point Montoring for Detection of DoS Attacks," *IEEE Trans. Dependable and Secure Computing*, vol. 1, no. 4, 2004, pp. 193-208.
5. S. Staniford, J.A. Hoagland, and J.M. McAlerney, "Practical Automated Detection of Stealthy Portscans," *J. Computer Security*, vol. 10, no. 1-2, 2002, pp. 105-136.
6. D.F. Jerding and J.T. Stasko, "The Information Mural: A Technique for Displaying and Navigating Large Information Spaces," *IEEE Trans. Visualization and Computer Graphics*, vol. 4, no. 3, 1998, pp. 257-271.
7. B. Shneiderman, "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations," *Proc. IEEE Symp. Visual Languages*, IEEE CS Press, 1996, p. 336.
8. M. Friendly, "Corrgrams: Exploratory Displays for Correlation Matrices," *American Statistician*, vol. 56, no. 4, 2002, pp. 316-324.
9. Y. Gao, Z. Li, and Y. Chen, "A DoS Resilient Flow-Level Intrusion Detection Approach for High-Speed Networks," to be published in *Proc. 26th Int'l Conf. Distributed Computing*, 2006.
10. J. McPherson et al., "Portvis: A Tool for Port-Based Detection of Security Events," *Proc. ACM Workshop Visualization and Data Mining for Computer Security,* (VizSEC/DMSEC), ACM Press, 2004, pp. 73-81.
11. X. Yin et al., "VisFlowConnect: Netflow Visualizations of Link Relationships for Security Situational Awareness," *Proc. ACM Workshop Visualization and Data Mining for Computer Security* (VizSEC/DMSEC), ACM Press, 2004, pp. 26-34.
12. K. Lakkaraju, W. Yurcik, and A. J. Lee, "NVisionIP: Netflow Visualizations of System State for Security Situational Awareness," *Proc. ACM Workshop Visualization and Data Mining for Computer Security* (VizSEC/DMSEC), ACM Press, 2004, pp. 65-72.
13. K. Abdullah et al., "IDS Rainstorm: Visualizing IDS Alarms," *Proc. IEEE VizSEC*, IEEE CS Press, 2005, pp. 1-10.

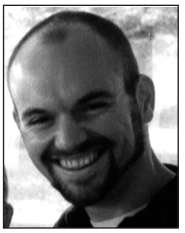
Pin Ren *is a PhD student in the Department of Electrical Engineering and Computer Science at Northwestern University. His research interests include computer graphics, information visualization, and scientific visualization. Ren has a BS in computer science from Peking University, China, and an MS in computer science from Northwestern University. Contact him at p-ren@cs.northwestern.edu.*

Yan Gao *is a PhD student in the Department of Electrical Engineering and Computer Science at Northwestern University. Her research interests include network security, network measurement, and monitoring. Gao has a BE in electrical engineering and an MS in system engineering, both from Xi'an Jiaotong University, China. Contact her at ygao@cs.northwestern.edu.*



Zhichun Li *is a PhD student in the Department of Electrical Engineering and Computer Science at Northwestern University. His research interests include network security, network measurement, and monitoring, and data streaming. Li has a BS in physics and an MS in computer science, both from Tsinghua University, China. Contact him at lizc@cs.northwestern.edu.*

Yan Chen *is an assistant professor in the Department of Electrical Engineering and Computer Science at Northwestern University. His research interests include network security, network measurement, peer-to-peer systems, and wireless and ad hoc networks. Chen has a PhD in computer science from the University of California, Berkeley. He won the Department of Energy Early Career award in 2005. Contact him at ychen@cs.northwestern.edu.*



Benjamin Watson *is an associate professor of computer science at North Carolina State University. His research interests include design graphics, considering computer imagery as and about designed objects; adaptive display; and the intersections between graphics and perception, design, and interaction. Watson has a PhD in Computer Science from Georgia Institute of Technology's Graphics, Visualization, and Usability Center. He is a senior member of the IEEE. Contact him at bwatson@ncsu.edu.*