

Network Traffic Analysis Using Behavior-Based Clustering

Kenneth Theriault, Daniel Vukelich, Wilson Farrell, Derrick Kong, John Lowry
BBN Technologies

theriault@bbn.com, vukelich@bbn.com, wfarrell@bbn.com, dkong@bbn.com, jlowry@bbn.com

Abstract

Conventional tools for network-level misuse and intrusion detection, signature-based detectors and anomaly detectors, are effective in many situations, but suffer from a number of drawbacks and limitations.. This paper describes a proof of concept study of a complementary analysis tool which avoids some of the pitfalls of these conventional tools, and should allow a human operator to reliably identify incidents deserving detailed attention; the intent is not to provide an alert, but rather to group or consolidate observed data, and to prioritize the groups for operator assessment. This methodology, termed behavior-based clustering, consolidates packet header data into clusters based on similarity of observed behavior, e.g., source IPs are clustered together based on their pattern of destination port usage. Cluster characteristics, or features, are then employed to sort the clusters into a prioritized list for assessment. This methodology is applied to two real world data sets: five hours of access point data, and seventy-two hours of enterprise gateway data from two different organizations. Clustering is found to greatly reduce the total volume of data to be reviewed; the number of clusters formed over a given time interval is found to grow slowly with time (as $\log(t)$); and, simple sorting rules are shown to be effective in prioritizing suspicious clusters for investigation.

1. Introduction and background

Detection of attacks, misuse, or generally suspicious behavior at the network traffic level is of interest in a variety of situations. Typically, this detection processing or analysis is carried out by using signature-based detectors or anomaly detectors to flag activities of interest for further investigation by a human operator; these methods are often augmented by algorithms which correlate or associate multiple events associated with the same suspicious activity. In principle, the operator's attention can then be focused on a limited number of consolidated alerts which represent valid indications of activities of concern.

Although both signature-based and anomaly detectors deal well with a number of significant activities of interest, this ideal vision is seldom achieved: operators are typically inundated with false alerts, and attacks or other behaviors often go undetected (or are lost in the background noise).

In large part, this result stems from the inherent properties of the anomaly and signature detectors. Signature-based detectors attempt to match the observed data (say, a stream of packet headers) to a known 'bad' template, the intrusion detection analog of a matched filter used in signal processing. Signature-based approaches suffer from a number of drawbacks:

- they cannot detect exploits whose pattern is not available a priori;
- they can be spoofed by an intelligent adversary who explicitly varies the character of an exploit, e.g., as in polymorphic viruses; and,
- they are not sensitive to partial matches, but rather produce only a binary decision if a complete match occurs.

Anomaly detectors attempt to identify variations in usage patterns, traffic levels, etc., which may be indicative of attacks or misuse; the variations are detected, in general, by comparing short-term statistics (such as means or medians) of the indicator variable with longer term statistics. An anomaly occurs when the short-term statistic diverges significantly from the long-term. Analogous detectors have been used for detection of transients in a variety of physical systems. Difficulties with anomaly detectors include:

- they can be trained by an intelligent adversary, who gradually increases the level of suspicious behavior so that the short-term statistic never exceeds the long term;

- they are sensitive to non-stationary behavior in the observed variable, such as diurnal variation in usage, or naturally-occurring (benign) fluctuations; and,
- they have limited sensitivity to very short-lived activities.

These limitations motivate the use of additional, complementary tools for analysis of network traffic. This paper addresses such an alternative, termed behavior-based clustering. This analysis concept avoids many of the underlying assumptions of signature-based and anomaly detectors, and appears to have value based on preliminary analysis of some real-world data.

2. Concept overview

In behavior-based clustering, traffic elements (packets) are grouped into clusters based on selected behavioral aspects (e.g., destination port usage by a given source IP), so that data with similar properties can be analyzed as a single entity. Each resulting cluster is characterized in terms of a set of descriptive features which summarize the behavior represented in the clusters; analysis or assessment of all the packets represented by a cluster can be performed on the basis of appropriately-selected features.

Since the clusters represent a large group of packets with identical (or very similar) behavior or properties, they can be analyzed as a single group, in principle effecting a substantial condensation or compression of the data. This approach does not inherently involve binary hypothesis testing, assumptions about traffic stationarity, or a priori knowledge of misuse patterns.

- Cluster features are used to sort or prioritize the clusters for examination; no 'hard', binary decision or thresholding is required.
- Clustering makes no assumptions about traffic stationarity or use patterns.
- No signature information is employed in the cluster formation process.

This initial concept study applied behavior-based clustering to two real-world data sets; the notional goal was to demonstrate the ability of clustering to aid in detection of probes, scans, and other suspicious behavior in high-volume data streams seen at an enterprise gateway and at a network access point. Issues to be addressed included

- the effectiveness of clustering to compress or consolidate the raw header data;
- the rate at which cluster count grew over time; and,
- the analytical value of the cluster features in scan, probe, and misuse detection.

The study was confined to analysis of header data in TCP SYN packets. The particular clustering approach employed for this problem was aggregation of source IPs (SIPs) which exhibited similar port usage behaviors, irrespective of the destination IPs (DIPs). The resulting clusters hence represent users with similar service usage profiles, which are sensibly analyzed as a group. This preliminary study did not attempt to identify a 'best' clustering algorithm (nor a best distance measure), but rather sought to validate the concept of behavior-based clustering.

Section 3 contains an analytical description of the clustering process, and, in particular, the distance measure used to form the clusters. Section 4 describes the data sets used in the analysis; the results of the study are given in Section 5, and the conclusions summarized in Section 6. Section 7 contains recommendations for further study and expansion of the behavior-based clustering concept.

3. Analysis

3.1 Data and notation

Let $p(sip, dip, dp)$ denote the joint probability of occurrence of source IP sip , destination IP dip , and destination port dp . This joint probability, sometimes termed the joint probability table, is computed by averaging over the packet header data in the observation (time) interval of interest. This computation was carried out on the test data using SPADE, a snort plugin. The resulting sample joint probability is used to characterize the data for the purposes of clustering.

The method used in this particular analysis is to cluster together SIPs which have similar DP usage patterns, without regard to the actual DPs involved, an approach which requires that the dependence on dp be removed from the joint probability function. Define the joint probability of source IP and destination port as

$$p'(sip, dp) = \sum_{dip} p(sip, dip, dp)$$

where the sum is taken over all destination IPs, and the dependence on dip is removed. Then the conditional probability of a DP for a given SIP is

$$p'(dp | sip) = \frac{p'(sip, dp)}{\sum_{dp} p'(sip, dp)}$$

where the denominator is the probability of occurrence of the specified sip .

3.2. Clustering algorithm

As discussed above, the objective of clustering is to group together all SIPs whose DP usage patterns are 'sufficiently' similar. Similarity is measured in terms of the distance between the two conditional probability distributions, $p'(dp/sip1)$ and $p'(dp/sip2)$; the actual distance measure employed is presented and discussed in Section 3.3.

A variety of approaches may be employed to perform the actual clustering operation, such as k-nearest neighbor and simulated annealing. This study employed brute-force clustering, in which the distance between all SIP pairs was computed, and the resulting distances used to form the clusters. In this approach, a SIP is assigned to a cluster if it is within a threshold distance of any member of the cluster. A SIP, which is less than a threshold distance from SIP members of two or more clusters serves to bridge or consolidate the clusters into a single new cluster.

The clustering algorithm proceeds as follows. Compute the distance between next unclustered SIP on the list and each of the SIPs which are already members of clusters. Any existing clusters which have elements less than a threshold distance away from the new SIP are merged, and the new SIP is joined to the merged cluster. If no existing cluster elements are within threshold distance of the new SIP, the new SIP is used to start a unique cluster. This process is repeated until all SIPs are members of clusters. Given n SIPs to be clustered, this process requires $n^2/2 - n$ comparisons.

Brute force was selected (over simulated annealing, which was initially used) because it provided consistent, deterministic results, and because it was, at the scales of interest, computationally competitive. Different clustering algorithms will likely change the details of the results, but should not change the general behavior or performance of this overall methodology.

3.3. Distance measure

The distance measure is used by the clustering algorithm to quantify the similarity between the DP usage of two SIPs, characterized as their conditional probability distributions, $p'(dp/sip1)$ and $p'(dp/sip2)$. If the distance between these two functions is sufficiently small, less than a specified threshold, $sip1$ and $sip2$ should fall into the same cluster. There are a large number of alternative distance measures which might reasonably be applied to this problem. This study employed as a distance measure a modified form of the mutual information between destination ports and the two source IPs to be compared.

The mutual information between destination port and source IP is given by

$$I(dp; sip) = \frac{1}{\sum_{sip1, sip2} \sum_{dp} p'(sip, dp)} \sum_{dp, sip1, sip2} p'(dp, sip) \log_2 \frac{p'(dp | sip)}{p'(dp)}$$

where

$$\begin{aligned}
p'(dp) &= p'(dp, sip1) + p'(dp, sip2) \\
&= p'(dp|sip1)p'(sip1) + p'(dp|sip2)p'(sip2)
\end{aligned}$$

The leading term is a scale factor which normalizes the joint probability to sum to unity over the region of interest. This measure is zero when the two conditional port use distributions are identical, and is symmetric in $sip1$ and $sip2$. This equation can be rewritten as

$$I(dp; sip) = C \int_{dp, sip1, sip2} p'(dp|sip)p'(sip) \log_2 \frac{p'(dp|sip)}{p'(dp|sip1)p'(sip1) + p'(dp|sip2)p'(sip2)}$$

where C represents the scale factor, and the expression now reveals that the result is dependent on $p'(sipn)$, the probability of occurrence of source IP n . This dependence was deemed inappropriate for the purposes of this analysis, as the objective is to determine if $p'(dp|sip1) = p'(dp|sip2)$ without regard to the prevalence of either source IP in the observation set.

To reduce the influence of the a priori probabilities of the source IPs to be compared, a modified version of the measure was employed, in which $p'(sip1)$ and $p'(sip2)$ were forced to 0.5, effectively removing any bias toward either source IP based on its representation in the measurement data set.

This modified mutual information can be written as

$$I_M(dp, sip) = \frac{1}{2} \int_{dp, sip1, sip2} p'(dp|sip) \log \frac{2 p'(dp|sip)}{p'(dp|sip1) + p'(dp|sip2)}$$

The constraints on $p'(sipn)$ reduce the original scale factor, C , to unity.

3.4. Cluster features

Each cluster was characterized by a set of features, selected for their conjectured value in assessing suspicious behavior, and their ability to inform the process of developing the analysis methodology. These features are listed in Table 1, together with a brief description.

Table 1. Cluster feature definitions

Feature	Description
clusterProb	Fraction of data set represented by this cluster
DomPort	Port with greatest usage, averaged over cluster
DomPortProb	Fraction of cluster port usage of DomPort
OtherProb	Fraction of usage of other ports, 1-DomProb
#ports	Total number of unique ports used by SIPs in the cluster
#dips	Total number of unique DIPs used by SIPs in the cluster
#sips	Total number of SIPs in the cluster
Entropy	Entropy of $p_c(dp)$
DistFromUni	Entropy distance between $p_c(dp)$ and a uniform distribution with the same number of ports,
AvgDistFromCentroid	Average distance of a cluster SIP from centroid ($p_c(dp)$)
MaxDistFromCentroid	Maximum distance of a cluster SIP from centroid ($p_c(dp)$)

Some of these features are based on the cluster's overall DP usage distribution, computed by averaging $p'(dp, sip)$ over the SIPs participating in the cluster:

$$p_c(dp) = \frac{\sum_{sip \in C} p'(dp, sip)}{\sum_{dp \in sip \in C} p'(dp, sip)}$$

Additional computed features are the entropy of the cluster port distribution,

$$H = - \sum_{dp} p_c(dp) \log_2(p_c(dp))$$

and the distance of the cluster port distribution from uniform, measured in terms of entropy,

$$H = \log_2(N_{dp}) - \sum_{dp} p_c(dp) \log_2(p_c(dp))$$

where N_{dp} is the number of unique ports in the cluster (the number of non-zero values of $p_c(dp)$).

Two features, AvgDistFromCentroid, and MaxDistFromCentroid, characterize the size of the cluster in terms of the clustering distance measure, relative to the centroid of the cluster. The cluster centroid is simply $p_c(dp)$, the cluster port distribution. The distance between the centroid and the port distribution of each of the participating SIPs is computed in terms of the modified mutual information measure used for clustering:

$$I_M(sip) = \frac{1}{2} \sum_{dp} p'(dp | sip) \log_2 \frac{2p'(dp | sip)}{p'(dp | sip) + p_c(dp)} + p_c(dp) \log_2 \frac{2p_c(dp)}{p'(dp | sip) + p_c(dp)}$$

The features are then

$$\text{AvgDistFromCentroid} = \sum_{sip \in C} I_M(sip) p_C(sip)$$

$$\text{MaxDistFromCentroid} = \max_{sip \in C} \{I_M(sip)\}$$

In addition to these features, the structure of each cluster (its component SIPs) was stored in a database to support detailed analysis and development.

4. Test data

The behavior-based clustering analysis concept was tested on two real-world data sets, an internet access point, and an enterprise gateway. This section gives a comparative overview of these two test data sets.

The access point data set comprises a total of five hours of data, beginning at 1:00 PM on a weekday afternoon, of which 3 hours 45 minutes are contiguous. This access point is subject to traffic which is asymmetrically routed, which complicates interpretation of the observations. Internal, presumably benign, users, as distinguished from external, generally suspect, users are known only approximately. Of this entire data set, about 35 minutes overlap the results from an independent analysis/alerting system, which provides a notional ground truth. In addition, manual analysis was performed on the clusters generated during this period.

The enterprise data set comprises over 72 hours of contiguous data, beginning at 3:00 PM on a weekday afternoon. The traffic is symmetrically routed, and the sets of 'inside' and 'outside' users are known with good confidence. At the time of this study, no independent assessment of this data set was available; hence, an independent manual analysis of the clustered data was performed, and used as the ground truth.

A summary of the properties of the two data sets is given in Table 2.

Table 2. Summary of test data set properties

Property	Gateway	Access point
Duration	73.5 hr	5 hr
TCP SYN packets	1.4×10^7	4.7×10^6
Unique (<i>sip</i> , <i>dip</i> , <i>dp</i>) triples	2.6×10^6	4.7×10^5
SIPs	1.2×10^5	3.4×10^4
DPs	2.1×10^4	4.1×10^3

Table 3 shows the fraction of SYN packets flowing in each direction. The very small fraction of internal to internal traffic in the gateway data could be due to the position of the collection point with respect to the participating 'internal' networks, or might also be due to incomplete or inaccurate identification of 'internal' addresses.

Table 3. Summary of test data directionality

	External to internal	Internal to external	Internal to internal
Gateway	0.45	0.55	0.0005
Access point	0.15	0.82	0.03

Another key distinction between the two data sets is the gross pattern of destination port usage: while both traffic sets are dominated by port 80/443 traffic, the access point data contains a greater fraction of port 80/443 data than does the gateway, as seen in Figure 2. The larger fraction of 'other' traffic in the gateway data reflects the much wider range of destination ports in that data set, as seen in Table 2.

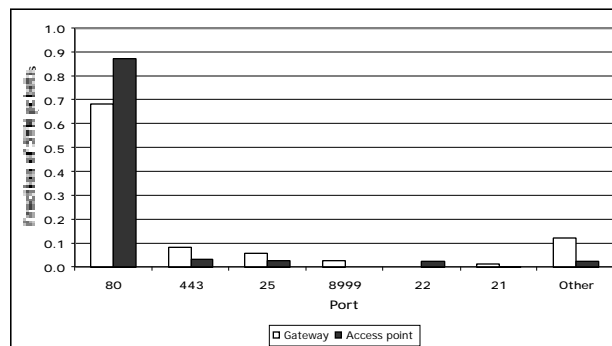


Figure 2. Port composition of data sets

5. Results

This section describes the results of applying behavior-based clustering analysis to the gateway and access point data sets described in Section 4. In order to achieve useful results, the header data subjected to clustering was first filtered to select inbound-only traffic, and to suppress predominant ports; this is discussed in Section 5.1. The compression or condensation achieved with this analysis approach is presented in Section 5.2; Section 5.3 describes growth in the number of clusters as the analysis interval increases, and Section 5.4 summarizes the analysis – behavior characterization – results achieved via clustering relative to ground truth.

5.1 Filtering

The data to be clustered were first filtered to select only inbound packets, and to exclude those whose DP was any of ports 80, 443, or 25. At a practical level, these steps were taken to reduce the mass of data in a sensible fashion. However, a focus of the investigation was to identify the incidence of scans and probes of 'inside' resources, which provided an operational motivation for application of clustering to inbound traffic only.

When clustering was attempted without suppression of predominant or heavily used ports, SIPs with high usage of these ports would 'attract' almost any SIP with even a low-probability interaction with one of these ports into the same cluster, despite possibly significant variations in usage of other ports. This effect was eliminated by removing events whose DP was one of the predominants: 80, 443, or 25. Operationally, this was equivalent to assuming that (absent DDoS attacks), SYN packets destined for one of these common ports were benign.

5.2 Compression

One of the motivations for investigating behavior-based clustering is its potential for data compression or condensation. Such compression, if it results in compact clusters which are descriptive of the underlying behavior, should reduce the amount of information to be reviewed by an analyst without reducing the analyst's ability to make a correct assessment of the underlying behavior. Figure 4 shows the compression achieved using clustering for both the gateway and access point data sets, as well as the effect of directional filtering, and suppression of predominant ports. Clustering reduces the volume of (filtered) event data by about three orders of magnitude for both data sets.

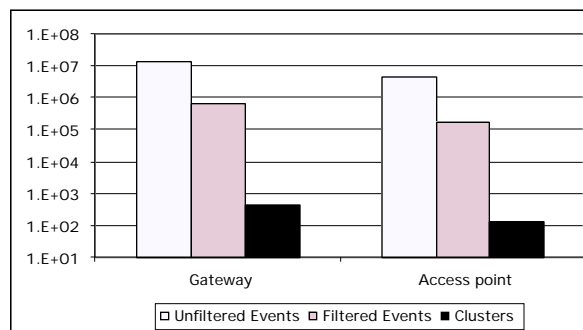


Figure 4. Data compression achieved with clustering

5.3 Cluster growth over time

One of the issues in the use of clustering is the rate of growth in the number of clusters over time: if the number of clusters grows linearly over time, as the observation interval increases, the burden placed on the analyst may become un-supportable even given the compression provided by the clustering process. Figure 5 shows the cumulative number of clusters formed from the access point data, for both unfiltered (upper curve) and filtered (lower curve) data sets, as a function of observation interval. The number of clusters grow logarithmically with time in both instances. This result is very encouraging, as logarithmic growth over time is very slow. Extrapolation of these 5 hours of data indicates that only 180 clusters would be generated for analysis over 24 hours.

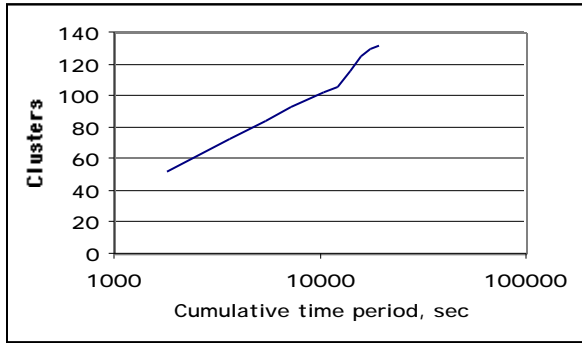


Figure 5. Cluster growth with time for access point data

Figure 6 shows cluster growth with time for the gateway data. These data cover a much longer period of time (three days) than the access point data (five hours), so that diurnal cycles of use can be clearly seen, with high growth rates during time periods from about 0600-1700 each day, and low growth for the balance of the day. Within each segment, growth is logarithmic in time; that is, the curves can be represented as piecewise linear in $\log(t)$.

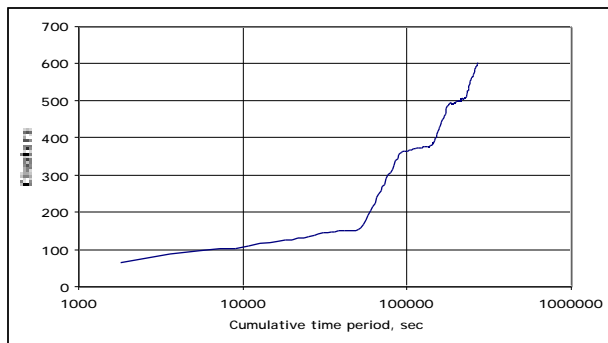


Figure 6. Cluster growth with time for gateway data

These growth results, obtained with data sets collected from two different types of observation points at different times, strongly suggest that piecewise $\log(t)$ growth of cluster count can be expected in a wide variety of situations.

5.4 Analysis results

The purpose of this analysis was to demonstrate the potential of behavior-based clustering and feature-based analysis to aid in the detection of misuse and suspicious activities using real world data. The cluster data from both data sets was analyzed by performing simple (single parameter) sorts on the cluster features, and comparing those clusters which sorted to the top of the resulting list with "truth" data derived from an independent assessment system (for the access point data) and from a manual analysis of the clusters (for both sets). The operational concept is that feature-based sorting would provide a reliable means for prioritizing a operator's workload. For this proof-of-concept study, no attempt was made to optimize or generalize the sorting rules; the goal was to identify one or more simple rules for each data set which gave appropriate priority to the clusters which 'the 'truth' results indicated were of greatest concern.

Manual analysis of the clusters involved examination of the underlying structure of the clusters (SIPS, specific ports used, etc.). Clusters were put into one of three behavior categories: benign, or low importance to investigate; moderate importance (cannot dismiss without further investigation), and high importance (must investigate).

Thirty-five minutes of access point data were analyzed; this segment was selected because it overlapped results from an independent, automated assessment system which provided an additional source of ground truth. Thirty-three clusters were formed from this data; the hand analysis results are summarized in Table 4.

Table 4. Manual cluster analysis results for access point data (35 min segment)

Importance	Clusters
Low	8
Moderate	15
High	8
Total	33

It was found that by sorting on the clusterProb feature (fraction of event data represented by the cluster), 6 of the 8 high-importance clusters sorted to the top of the cluster list; all 8 sort with the top 20. Further, the clustering analysis proved to be more sensitive than the In addition, clustering identified a broad port scan and a known Trojan that had been missed by the automated analysis tool.

Analysis of the clusters from the gateway data produced similarly encouraging results. No 'automated' ground truth was available for this data set, so assessment was based solely on comparison of the cluster sorting results with the manual analysis. A total of 604 clusters were formed from the 73.5 hours of header data; the results of the manual analysis of these clusters is summarized in Table 5.

Table 5. Manual cluster analysis results for gateway data

Importance	Clusters
Low	547
Moderate	43
High	14
Total	604

Two sorts were found to be effective in prioritizing the high importance clusters. By sorting on the feature #ports (total number of unique ports used by SIPs in the cluster) 9 of the 14 high importance clusters sort within the top 45 entries in the cluster list. By sorting on #dips (total number of unique DIPs used by SIPs in the cluster), the remaining five high-importance clusters sort to the top 7 entries in the cluster list.

6. Conclusions

The results presented above demonstrate the utility of behavior-based clustering for data compression and the effectiveness of cluster features in identifying behaviors of interest or concern. Further, cluster count scales very favorably with time, growing as $\log(t)$ for both the gateway and access point data sets. This growth law appears to hold in a piecewise sense even for the gateway data, where the underlying exhibits diurnal non-stationarities. Although conducted over a limited data set using experimental algorithms, these results appear to validate the potential utility of behavior-based clustering as a network traffic analysis tool.

Behavior-based clustering should be considered as an additional tool for analysis of network traffic: it is a complement to, and not a substitute for conventional signature-based detectors, rules for identification of known bad ports or suspicious SIPs, etc. It provides an alternative view of the observed data which should allow an operator to identify inherently unusual use patterns in an efficient way.

7. Further investigations

This study, although providing a preliminary validation of the behavior-based clustering concept, also revealed a number of sensitivities and issues which must be addressed before this methodology can be applied operationally. Selection of an appropriate distance measure is an ongoing concern. Despite the use of the modified distance measure described in Section 3.3, cluster formation is still affected by the presence of dominant ports: SIPs which use the same port very frequently will tend to cluster together, despite complete disparities in their use of other ports. This effect is due to the weighted nature of the distance metric, in which high-usage ports tend to drive the match. This could be addressed by further port sup-

pression (as was done with ports 80, 443, and 25 in the analysis examples), or by use of an alternative distance measure, such as port overlap (the number of common ports without respect to usage levels), or the summed absolute difference.

Port overlap is motivated by the notion that variations in the relative level of useage of ports are not significant, e.g., relative useage of ports 80 and 443; rather, what is significant is any activity by one source IP on a port untouched by the other source IP. Let $D1$ and $D2$ be the sets of non-zero-probability destination ports for $sip1$ and $sip2$ respectively. Let $size(D)$ be the number of element in set D . Then the port overlap measure is given by

$$D_O = size(\cup\{D1, D2\} - \cap\{D1, D2\}).$$

A value of 0 indicates the two source IPs are using the same set of ports.

The summed absolute difference is analogous to an L1 norm; it comprises the summed absolute differences between the two conditional port usage probabilities:

$$D_S = \sum_{dp} |p'(dp | sip1) - p'(dp | sip2)|$$

This measure lies in the range [0,2], with 0 representing a perfect match, and 2 representing a complete mismatch (no common ports). This requires no normalization, as the probabilities themselves already sum to unity; in this regard, the summed absolute difference is a 'natural' distance measure. The effect of a threshold applied to this measure is easy to visualize and interpret.

Various extensions of behavior-based clustering suggest themselves:

- application to other behaviors, such as clustering destination IPs by destination port usage, hence profiling how one's own resources are actually being used;
- application to packet types other than TCP SYN; and,
- extension to stateful analysis of TCP connections.

8. Acknowledgements

This work was sponsored by the Advanced Technology Office of the Defense Advanced Research Projects Agency, under the Operational Partners in Experimentation Program, Mr. Brian Witten Program Manager, Contract Number F30602-98-C-0012.